Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – K141502

KLASIFIKASI PERINTAH BAHASA NATURAL MENGGUNAKAN GLOBAL VECTORS FOR WORD REPRESENTATIONS (GLOVE), CONVOLUTIONAL NEURAL NETWORKS, DAN TEKNIK TRANSFER LEARNING PADA APLIKASI CHATBOTS

IRFAN HANIF
NRP 05111440000177

Dosen Pembimbing
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – K141502

KLASIFIKASI PERINTAH BAHASA NATURAL MENGGUNAKAN GLOBAL VECTORS FOR WORD REPRESENTATIONS (GLOVE), CONVOLUTIONAL NEURAL NETWORKS, DAN TEKNIK TRANSFER LEARNING PADA APLIKASI CHATBOTS

IRFAN HANIF
NRP 05111440000177

Dosen Pembimbing
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

FINAL PROJECT – K141502

NATURAL LANGUAGE COMMAND SENTENCE CLASSIFICATION USING GLOBAL VECTORS FOR WORD REPRESENTATIONS (GLOVE), CONVOLUTIONAL NEURAL NETWORKS, AND TRANSFER LEARNING FOR CHATBOTS

IRFAN HANIF
NRP 05111440000177

Supervisors
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KLASIFIKASI PERINTAH BAHASA NATURAL MENGGUNAKAN GLOBAL VECTORS FOR WORD REPRESENTATIONS (GLOVE), CONVOLUTIONAL NEURAL NETWORKS, DAN TEKNIK TRANSFER LEARNING PADA APLIKASI CHATBOTS

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:
IRFAN HANIF
NRP: 05111440000177

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Agus Zainal Arifin, S.Kom., M.Kom.
NIP: 19720809 199512 1 001 ..... (Pembimbing 1)

Dini Adni Navastara, S.Kom., M.Sc.
NIP: 19851017 201504 2 001 ..... (Pembimbing 2)

SURABAYA
MEI 2018

[Halaman ini sengaja dikosongkan]

KLASIFIKASI PERINTAH BAHASA NATURAL MENGGUNAKAN GLOBAL VECTORS FOR WORD REPRESENTATIONS (GLOVE), CONVOLUTIONAL NEURAL NETWORKS, DAN TEKNIK TRANSFER LEARNING PADA APLIKASI CHATBOTS

Nama Mahasiswa : Irfan Hanif
NRP : 05111440000177
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRAK

Akhir-akhir ini, chatbots menjadi salah satu alternatif yang sangat menjanjikan dalam menangani Frequently Asked Question (FAQ) yang diajukan konsumen-konsumen kepada organisasi atau perusahaan. Kini, ketidakmampuan chatbots dalam mengatasi keragaman kata yang digunakan para konsumennya menjadi masalah karena membuat mereka mulai frustasi dalam menggunakan chatbots tersebut. Maka dari itu, diusulkan metode klasifikasi perintah bahasa natural menggunakan Global Vectors for Word Representations (GloVe), Convolutional Neural Networks, dan teknik Transfer Learning pada aplikasi chatbots. Metode ini diusulkan karena kemampuannya yang bagus dalam membuat sistem chatbots mengenali makna kata, mengelompokkan kalimat-kalimat yang sering diajukan berdasarkan jawabannya, dan mengatasi permasalahan perusahaan yang tidak mempunyai data yang cukup untuk dipelajari chatbots. Ketiga algoritma ini terbukti memberikan hasil yang sangat baik untuk menjadi sistem inti chatbot karena hasil akhir uji coba model berhasil mencapai nilai F-Measure sebesar 95,6% dengan menggunakan data training yang relatif sedikit.

Kata kunci: chatbot, klasifikasi, GloVe, convolutional neural networks, transfer learning

[Halaman ini sengaja dikosongkan]

NATURAL LANGUAGE SENTENCE COMMAND CLASSIFICATION USING GLOBAL VECTORS FOR WORD REPRESENTATIONS (GLOVE), CONVOLUTIONAL NEURAL NETWORKS, AND TRANSFER LEARNING FOR CHATBOTS

Student Name	: Irfan Hanif
NRP	: 05111440000177
Departement	: Informatika FTIK-ITS
Supervisor 1	: Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Supervisor 2	: Dini Adni Navastara, S.Kom., M.Sc.

ABSTRACT

Lately, chatbots have become one of the most promising alternatives in dealing with Frequently Asked Questions (FAQs) that consumers propose to organizations or companies. Now, the inability of chatbots to cope with the diversity of words used by its customers becomes a problem because it makes them start frustrating in using chatbots. Therefore, this final project proposed a natural language sentence classification method using Global Vectors for Word Representations (GloVe), Convolutional Neural Networks, and Transfer Learning for chatbots application. This method is proposed because its remarkable ability to make chatbots system recognize the meaning of the word, classifying FAQs sentences based on the answer, and overcoming the problem of companies that do not have enough data for chatbots to learn. These three alogithms proved to provide excellent results for being the core of chatbots system because the final test results of the model achieved F-Measure of 95,6% using relatively few training data.

Keywords: chatbot, classification, glove, convolutional neural networks, transfer learning

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur saya ucapkan kepada Allah Yang Maha Kuasa. Karena atas karunia serta rahmat-Nya saya dapat menyelesaikan tugas akhir yang berjudul:

KLASIFIKASI PERINTAH BAHASA NATURAL MENGGUNAKAN *GLOBAL VECTORS FOR WORD REPRESENTATIONS* (GLOVE), *CONVOLUTIONAL NEURAL NETWORKS,* DAN *TEKNIK TRANSFER LEARNING* PADA APLIKASI *CHATBOTS*

Sesungguhnya dalam pengerjaan dan penulisan tugas akhir ini, saya merasa sangat antusias sekali dalam menyelesaikan dan melalui keseluruhan prosesnya. Tidak lain karena topik yang saya ambil ini merupakan bagian penting yang tidak terpisahkan dari Revolusi Industri 4.0, yaitu teknologi kecerdasan buatan. Oleh karena itu, saya berharap sekali pada institusi tempat saya mengenyam pelajaran hidup, Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS), untuk gencar membawa seluruh elemen akademisinya mengambil bagian menjadi penggerak utama dalam revolusi industri keempat ini.

Melalui kesempatan ini pula, saya ingin berterima kasih kepada seluruh pihak yang telah mendukung saya dalam menyelesaikan tugas akhir ini. Khususnya kepada Bapak dan Ibu saya yang selalu tulus dan tanpa pamrih mendukung saya dalam bentuk apapun. Pengorbanan kalian tak akan pernah saya lupakan seumur hidup saya. Ucapan terima kasih juga saya berikan kepada Pak Agus dan Bu Dini yang sudah sangat baik dalam membimbing saya. Termasuk memberi nasihat, mengorbankan waktu, tenaga, serta pikiran-pikirannya untuk saya. Terakhir, saya memberikan rasa hormat saya sedalam-dalamnya kepada seluruh dosen, guru, dan teman-teman saya seperjuangan.

Akhir kata, saya berharap semoga hasil dari tugas akhir ini, baik berupa hasil penelitian maupun ide dan buah pikiran yang menyertainya, dapat memberikan kontribusi berarti tidak hanya untuk bangsa ini, tetapi juga pada kehidupan umat manusia seluruhnya.

Surabaya, Mei 2018

Irfan Hanif

DAFTAR ISI

LEMBAR PENGESAHAN vii
ABSTRAK ix
ABSTRACT xi
KATA PENGANTAR xiii
DAFTAR ISI xv
DAFTAR GAMBAR xxi
DAFTAR TABEL xxiv
DAFTAR PSEUDOCODE xxvi
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan 4
1.5 Manfaat 4
1.6 Metodologi 4
BAB II DASAR TEORI 13
2.1 Chatbots 13
2.2 Global Vectors for Word Representations (GloVe) 14
2.3 Transfer Learning 20
2.4 Convolutional Neural Networks (CNN) 23
2.5 Adaptive Subgradient Methods (AdaGrad) 27
2.6 Adaptive Moment Estimation (Adam) 28
2.7 Cosine Similarity 29

BAB III DESAIN 31
3.1 Desain Aplikasi GloVe 31
3.1.1. *Text Preprocessing* 31
3.1.2. Membangun *Word-Word Co-Occurrence Matrix* 33
3.1.3. *AdaGrad Training* 33
3.2 Desain Aplikasi CNN 34
3.2.1. *Text Preprocessing* 35
3.2.2. *Sentence Mapping (Average)* 35
3.2.3. Pemisahan Dataset 36
3.2.4. CNN *Learning* 36
3.2.5. Prediksi 37
3.2.6. Evaluasi 37
3.3 Desain Aplikasi Uji Coba GloVe dan CNN 38
3.3.1. GloVe dan CNN Tanpa Transfer Learning 38
3.3.2. GloVe, CNN, dan Transfer Learning Skenario 1 38
3.3.3. GloVe, CNN, dan Transfer Learning Skenario 2 39
3.4 Arsitektur CNN 41
3.4.1. Eksperimen Arsitektur CNN 41
3.4.2. Arsitektur CNN Pada Dataset LAPOR! 44
3.4.3. Arsitektur CNN Pada Dataset FAQs Schematics 45
3.4.4. Arsitektur CNN pada Transfer Learning Skenario1 45
3.4.5. Arsitektur CNN pada Transfer Learning Skenario2 46
3.5 Desain Aplikasi Chatbot 47

3.6 Desain *User Interface* Pada *Chatbot* 48
3.6.1. Chat 50
3.6.2. Pertanyaan Langsung (*Direct Question*) 50
3.6.3. *Multiple Choices of Direct Question* 51
3.6.4. Jawaban 53
BAB IV IMPLEMENTASI 55
4.1 Tools 55
4.2 Tahap-Tahap Implementasi 55
4.3 Mendapatkan Word Embeddings dengan GloVe 56
4.3.1. Korpus 56
4.3.2. *Text Preprocessing* Pada Korpus 57
4.3.3. Membangun *Word-Word Co-Occurrence Matrix* 58
4.3.4. AdaGrad *Training* 58
4.4 Implementasi Pengujian GloVe Word Embeddings 61
4.4.1. Uji Fungsionalitas Semantik GloVe Word Embeddings 61
4.4.2. Uji 10 Kata Terdekat GloVe Word Embeddings. 62
4.5 Implementasi CNN 63
4.5.1. Text Preprocessing 63
4.5.2. Sentence Mapping and Averaging 64
4.5.3. Memisahkan Dataset 65
4.5.4. CNN Learning 66
4.5.5. Prediksi dan Evaluasi 66

4.6 Implementasi GloVe dan CNN Tanpa Transfer Learning 67
4.7 Implementasi GloVe, CNN, dan Transfer Learning Skenario 1 68
4.8 Implementasi GloVe, CNN, dan Transfer Learning Skenario 2 69
4.9 Implementasi Aplikasi *Chatbot* 71
4.9.1. Menjawab Direct Questions 71
4.9.2. Menjawab Dengan Prediksi CNN dan Menangani Ambiguitas 72
BAB V UJI COBA DAN EVALUASI 75
5.1 GloVe Word Embeddings 75
5.1.1 Deskripsi Korpus 75
5.1.2 Uji Fungsionalitas Semantik GloVe Word Embeddings 75
5.1.3 Uji dan Evaluasi 10 Kata Terdekat 78
5.2 GloVe dan CNN 79
5.2.1 Dataset 79
5.2.2 Uji dan Evaluasi Kemampuan Prediksi Model 81
5.3 GloVe, CNN, dan Transfer Learning 82
5.3.2 Dataset 83
5.3.3 Uji dan Evaluasi Kemampuan Prediksi Model 84
5.3.4 Uji dan Evaluasi Kecepatan Belajar Model 86
BAB VI KESIMPULAN DAN SARAN 91
6.1 Kesimpulan 91

6.2 Saran 92
DAFTAR PUSTAKA 93
LAMPIRAN 97
BIODATA PENULIS 147

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1.1. Confusion Matrix. 9
Gambar 1.2. Ilustrasi confusion matrix pada 3 kelas. 12
Gambar 2.1. Veronika, chatbots asal Indonesia 14
Gambar 2.2. Perbedaan proses learning pada (a) traditional machine learning dan (b) transfer learning (Pan et al., 2010). 20
Gambar 2.3. Cara kerja sebuah Convolution Layer. 25
Gambar 2.4. ReLU dan Tanh pada CIFAR-10 (Kryzhevsky et al., 2012) 26
Gambar 2.5. Perbedaan arsitektur Neural Network dengan Dropout (Srivastava et al., 2014) 27
Gambar 3.1. Desain aplikasi GloVe. 31
Gambar 3.2. Contoh matriks word-word co-occurrence. 32
Gambar 3.3. Desain aplikasi CNN. 34
Gambar 3.4. Mendapatkan Sentence Embeddings pada salah satu dokumen. 35
Gambar 3.5. Desain GloVe, CNN, dan Transfer Learning Skenario 1 38
Gambar 3.6. Desain GloVe, CNN, dan Transfer Learning Skenario 2 40
Gambar 3.7. Arsitektur CNN secara umum yang terdiri dari (a) Sentence Embeddings, (b) convolutional layer, (b) hidden layer, dan (c) output layer 42
Gambar 3.8. Arsitektur CNN yang digunakan pada dataset LAPOR! 44
Gambar 3.9. Arsitektur CNN yang digunakan pada dataset FAQs Schematics dengan output layer 22. 45
Gambar 3.10. Arsitektur CNN yang digunakan pada Transfer Learning Skenario 1. Layer konvolusi (berwarna biru) merupakan hasil transfer dari CNN yang dilatih dengan data LAPOR! 46

Gambar 3.11. Arsitektur CNN yang digunakan pada Transfer Learning Skenario 2. Layer konvolusi (berwarna biru) merupakan hasil transfer dari CNN yang dilatih dengan data LAPOR! sedangkan layer hidden dan layer output (berwarna hijau) merupakan hasil transfer dari CNN yang dilatih dengan data FAQs Schematics. 47
Gambar 3.12. Alur komunikasi antar end-point pada aplikasi chatbots 48
Gambar 3.13. Flowchart sistem bot dalam menangani chat. 49
Gambar 3.14. Desain user interface chat pada LINE Messenger. 50
Gambar 3.15. Desain antarmuka multiple choices of direct questions. Terlihat pada chat selanjutnya, pengguna memilih direct question "Contoh Soal NLC". 52
Gambar 4.1. Vektor kata sebesar 300 dimensi dan window sebesar 10 memberikan hasil yang paling baik (Pennington et al., 2014). 61
Gambar 5.1. Ilustrasi salah satu data sintetis untuk uji semantik GloVe Word Embeddings. 76
Gambar 5.2. Confusion matrix dari model GloVe dan CNN tanpa transfer learning. 81
Gambar 5.3. Nilai loss per epochs pada model tanpa transfer learning 82
Gambar 5.4. Nilai loss per epochs pada Skenario Transfer Learning 1 86
Gambar 5.5. Nilai loss per epochs pada Skenario Transfer Learning 2 87
Gambar 5.6. Kecepatan belajar tiap model ditinjau dari nilai loss per epochs 88
Gambar 5.7. Confusion matrix dari model Transfer Learning Skenario 1 89

Gambar 5.8. Confusion matrix dari model Transfer Learning Skenario 2.....90

[*Halaman ini sengaja dikosongkan*]

DAFTAR TABEL

Tabel 2.1. Contoh ratio kemungkinan kemunculan kata. 16
Tabel 2.2. Istilah yang digunakan pada Transfer Learning. 21
Tabel 2 3. Perbedaan teknik pada Transfer Learning. 22
Tabel 4.1. tools yang digunakan pada tugas akhir. 55
Tabel 5.1. Penggalan hasil uji coba fungsionalitas semantik GloVe Word Embeddings. Hasil selengkapnya terlampir pada Lampiran 2. 77
Tabel 5.2. Hasil uji coba 10 kata terdekat pada kata "seragam", "lokasi", dan "kontestan". 79
Tabel 5.3. Contoh beberapa data pada FAQs Schematics. Selengkapnya pada Lampiran 3. 80
Tabel 5.4. Tabel keterangan kelas pada confusion matrix. 83
Tabel 5.5. Beberapa contoh data pada dataset LAPOR!. 84
Tabel 5.6. Perbandingan $\mathcal{T}_{\mathcal{T}}$ sebelum Transfer Learning, Transfer Learning Skenario (1), dan Transfer Learning Skenario (2). 85

[*Halaman ini sengaja dikosongkan*]

DAFTAR PSEUDOCODE

Pseudocode 4.1. Proses parsing korpus Wikipedia Bahasa Indonesia. 56
Pseudocode 4.2. Proses dokumen-dokumen LAPOR! menjadi korpus. 57
Pseudocode 4.3. Kode text preprocessing pada korpus. 58
Pseudocode 4.4. Kode membangun matriks word-word co-occurrence. 59
Pseudocode 4.5. AdaGrad training pada GloVe. 60
Pseudocode 4.6. Kode uji fungsionalitas semantik GloVe Word Embeddings. 62
Pseudocode 4.7. Kode untuk mencari 10 kata terdekat dalam ruang spasial Word Embeddings dari kata tertentu. 63
Pseudocode 4.8. Proses text preprocessing setiap dokumen pada dataset. 64
Pseudocode 4.9. Proses mengubah dokumen menjadi Sentence Embeddings. 65
Pseudocode 4.10. Implementasi prediksi dan evaluasi model CNN. 66
Pseudocode 4.11. Implementasi kode GloVe dan CNN Tanpa Transfer Learning. 67
Pseudocode 4.12. Implementasi Source Task TS pada Transfer Learning Skenario 1. 68
Pseudocode 4.13. Implementasi Target Task pada Transfer Learning Skenario 1. 69
Pseudocode 4.14. Implementasi Target Task pada Transfer Learning Skenario 2. 70
Pseudocode 4.15. Mengecek apakah sebuah chat adalah direct question. 71
Pseudocode 4.16. Implementasi sistem dalam mengembalikan jawaban berdasarkan kelas pertanyaan. 72

Pseudocode 4.17. Chatbot memberikan respons multiple choice atau jawaban. 73
Pseudocode 4.18. Proses prediksi jawaban pada sistem chatbot. 73

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemampuan *chatbots* dalam memberikan sentuhan personal dengan lawan bicaranya dianggap memberikan nuansa baru dalam dunia Interaksi Manusia dan Komputer (*Human Computer Interface*). Konsep ini disebut *conversational interface.* ELIZA (Weizenbaum, 1966) adalah *chatbot* yang diciptakan pertama kali pada tahun 1966 (Abu Shawar et al, 2007). Tujuan sederhananya saat itu adalah untuk mengetahui apakah ELIZA mampu menipu manusia sehingga manusia merasa sedang berkomunikasi dengan manusia sungguhan. Singkat cerita, penelitian ini pun berlanjut dan menghasilkan *chatbot* lain yang bernama ALICE (Wallace, 2003). Kemampuan ALICE untuk berkomunikasi layaknya manusia sungguhan telah membuatnya memenangi kejuaraan Loebner (kejuaraan *chatbots* yang melombakan berapa lama *chatbots* dapat menipu manusia sungguhan) pada tahun 2000, 2001, dan 2004.

Contoh lain yang paling fenomenal adalah Siri yang diciptakan Apple pada tahun 2011. Siri merupakan *chatbot* yang dapat berkomunikasi dengan perintah suara untuk membuat *reminder*, jadwal janjian, menemukan tempat terdekat, dan lainnya. *Chatbot* telah menjelma menjadi alternatif untuk menggantikan pekerjaan yang melibatkan percakapan dengan pengguna di mana pekerjaan ini adalah pekerjaan sederhana yang sifatnya repetitif.

Schematics adalah *event* yang diselenggarakan oleh Himpunan Mahasiswa Teknik Computer-Informatika Institut Teknologi Sepuluh Nopember (HMTC ITS). Sebagai *event* yang menargetkan ribuan peserta Sekolah Menengah Atas (SMA), pihak panitia seringkali mendapatkan pertanyaan-pertanyaan seputar *event* yang mereka selenggarakan. Hal ini menjadi masalah ketika pertanyaan ini seringkali berulang dan ditanyakan ratusan kali setiap harinya. Demi menjaga *engagement* pada peserta-pesertanya, panitia menugaskan beberapa orang untuk menangani pertanyaan-pertanyaan ini. Permasalahan ini tidak hanya terjadi pada

Schematics. *Frequently Asked Questions* (*FAQ*) bukan barang baru pada perushaan/institusi yang menawarkan barang dan jasa. Sifatnya yang masif dan repetitif sesungguhnya tidak efektif dan efisien jika dikerjakan oleh manusia.

Beberapa solusi yang pernah dilakukan adalah dengan membuat mesin penjawab telepon otomatis atau FAQ berbasis web. Kelemahan dari mesin penjawab telepon adalah ketidakmampuannya dalam menjawab pertanyaan spesifik. Kebanyakan dari tugas mesin ini pun adalah meneruskan keluhan pengguna ke *customer service*. FAQ berbasis web mempunyai kelemahan yang cukup merepotkan. Pengguna harus menulusuri satu-satu pertanyaan tersedia yang sesuai dengan pertanyaan yang ingin diketahui jawabannya.

Chatbot pun akhirnya menjadi alternatif dari permasalahan ini setelah diketahui mempekerjakan manusia pada pekerjaan sederhana yang memiliki repetisi tinggi sangat tidak efisien. Hanya saja sistem ini masih sulit untuk digunakan karena kebanyakan *chatbot* yang dibangun tidak dibekali kemampuan untuk memahami kalimat percakapan. Contohnya ketika terjadi hal seperti ini:

Bot : Mohon pilih salah satu opsi di atas
Pengguna : Berapa biaya pendaftarannya?
Bot : Mohon pilih salah satu opsi di atas
Pengguna: Berapa jumlah uang yang harus dibayar?
Bot : Mohon pilih salah satu opsi di atas
Pengguna: *Meninggalkan ruang *chat**

Kenyataannya, pengguna mengharapkan *chatbots* dapat memahami bahasa yang digunakan sehari-hari. Sementara kebanyakan *chatbot* yang ada saat ini masih meminta masukan khusus seperti "`/biaya pendaftaran`" atau "`/info timeline`" agar dapat dimengerti oleh sistem (biasanya dengan tambahan karakter garis miring sebagai karakter khusus agar dapat diidentifikasi sebagai sebuah masukan). Kesalahan satu karakter saja akan membuat sistem tidak mengerti. Cara ini juga digunakan dengan alasan bahwa perusahaan seringkali tidak menyimpan

daftar pertanyaan pada FAQ sehingga hanya mampu mengira-ngira kombinasi pertanyaan-pertanyaan ini (ini yang membuat data latih awal untuk *chatbot* hanya sedikit). Maka, dibuatlah sintaks sederhana yang mewakili keragaman kombinasi kalimat-kalimat ini.

Permasalahan ini dapat diatasi dengan sistem *chatbot* yang mampu memahami berbagai macam kombinasi kalimat yang biasa ditanyakan (FAQ) dalam bahasa sehari-hari. Sistem *chatbot* harus mampu memahami makna kata dan memprediksi jawaban secara otomatis untuk berbagai macam pertanyaan dan permintaan dari pengguna dengan baik meskipun data latih awal yang ada relatif sedikit. Oleh karena itu, dalam tugas akhir ini bertujuan untuk mengembangkan sistem yang dapat memberikan jawaban secara otomatis berdasarkan perintah pengguna dalam bahasa natural menggunakan *Global Vectors for Word Representations* (GloVe), *Convolutional Neural Networks* (CNN), dan teknik *Transfer Learning.*

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana agar *chatbot* dapat mengenali makna dari kata?
2. Bagaimana agar *chatbot* dapat melakukan klasifikasi kombinasi-kombinasi kalimat dan dapat memprediksi kombinasi kalimat lain dengan akurat?
3. Bagaimana mengatasi *data training* yang sedikit?

1.3 Batasan Masalah

Batasan masalah pada tugas akhir ini antara lain:

1. Bahasa yang digunakan adalah Bahasa Indonesia.
2. Kata yang dapat digunakan adalah kata yang terdapat pada korpus. Kata singkat atau salah ketik yang tidak terdapat pada korpus juga tidak akan dikenali. Korpus yang digunakan adalah Wikipedia Bahasa Indonesia.
3. *Chatbot* yang dibangun adalah *closed-domain bot* yang artinya percakapan *chatbot* hanya dapat menjawab pertanyaan yang

sudah ditentukan sebelumnya sebelum *chatbot* masuk ke dalam tahap *learning*.

4. *Chatbot* hanya dapat menangani satu level percakapan. *Chatbot* tidak mampu menjawab pertanyaan berdasarkan percakapan sebelumnya.

1.4 Tujuan

Tugas akhir ini bertujuan untuk mengembangkan sistem yang dapat memberikan jawaban secara otomatis berdasarkan perintah pengguna dalam bahasa natural menggunakan *Global Vectors for Word Representations* (GloVe), *Convolutional Neural Networks* (CNN), dan teknik *Transfer Learning.*

1.5 Manfaat

Manfaat dari tugas akhir ini adalah pengguna dapat mendapatkan jawaban atas pertanyaan yang sering ditanyakan secara otomatis dengan menggunakan kalimat yang tidak baku atau tersetruktur, tetapi sistem dapat memahami maksud dari pengguna tersebut. Dengan begitu, perusahaan atau organisasi dapat meningkatkan *user engagement* karena membuat komunikasi antar konsumen dan perusahaan/organisasi menjadi lebih natural.

Manfaat lain yang bisa didapatkan dari perusahaan yang menggunakan sistem ini adalah efisiensinya yang cukup tinggi. Sistem ini lebih efisien daripada menggunakan manusia karena *chatbot* adalah mesin yang dapat hidup 24 jam non-stop tanpa lelah sehingga dapat menggantikan manusia pada pekerjaan sederhana yang repetitif sehingga tenaga manusia dapat melakukan pekerjaan yang lebih berdampak tinggi.

1.6 Metodologi

Langkah-langkah yang harus ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. **Penyusunan proposal tugas akhir**

Proposal tugas akhir ini berisi tentang penjelasan mengenai pendahuluan dari tugas akhir yang dibuat. Pendahuluan ini terdiri dari hal-hal yang melatarbelakangi tugas akhir, rumusan masalah

yang diangkat, batasan masalah yang ada, tujuan, dan manfaat dari tugas akhir ini. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi dalam pengerjaan tugas akhir ini.

2. Akuisisi Data

Data yang digunakan pada tugas akhir ini ada dua dataset. Dataset yang pertama adalah data kalimat yang pernah diajukan oleh para peserta Schematics. Data ini sudah terlabeli berdasarkan kategori jawabannya masing-masing. Pada tugas akhir ini, dataset ini didapatkan dengan cara melakukan wawancara kepada Kepala Divisi Humas Schematics 2017. Dataset ini selanjutnya akan disebut *Target Dataset* yang dilambangkan dengan ($\mathcal{D}_T$). Dataset yang kedua adalah data laporan masyarakat Indonesia yang terekam dan dipublikasi online di *https://data.go.id*. Data-data dokumen pada dataset ini sudah terlabeli menurut kategori laporan yang masuk. Dataset ini kemudian akan disebut *Source Dataset* yang dilambangkan dengan ($\mathcal{D}_S$). Syarat dari $\mathcal{D}_S$ haruslah mempunyai jumlah data yang relatif besar karena dataset ini akan menjadi data awal sebelum proses *Transfer Learning*. $\mathcal{D}_S$ juga harus mempunyai keidentikan dengan $\mathcal{D}_T$. Jika tidak, proses *Transfer Learning* tidak akan sukses (Semwal et al, 2010).

Kedua dataset ini digunakan untuk tujuan yang berkaitan. Dalam penjelasan yang sederhana, $\mathcal{D}_S$ digunakan menjadi input dalan CNN dan menghasilkan sebuah model. *Knowledge* dari model ini kemudian ditransfer (diambil sebagian dan dibangun kembali) untuk dilatih kembali dengan $\mathcal{D}_T$.

3. Studi Literatur

Pada studi literatur tugas akhir ini telah dipelajari sejumlah referensi yang relevan terhadap tugas akhir yang telah dikerjakan. Secara garis besar, ada tiga metode/teknik/algoritma yang menjadi pilar dalam tugas akhir kali ini. Yaitu, *Global Vectors for Word Representation* (GloVe), *Convolutional Neural Networks* (CNN), dan teknik *Transfer Learning*.

Salah satu teknik *term weighting* yang telah dipelajari selama dua dekade terakhir ini adalah *term frequency-inverse document frequency* atau lazim disebut TF-IDF. TF-IDF adalah salah satu teknik *count-based* model yang memberikan matriks *co-occurrence* sebagai hasil akhirnya. TF-IDF pernah digunakan sebagai bagian dari metode untuk mengukur kemiripan dokumen berita dan memberikan hasil yang relatif sangat baik (Arifin dan Setiono, 2002). Hanya saja, TF-IDF ternyata masih belum mampu mengenali hubungan antar kata dan memberikan masalah lain ketika sistem harus meningkatkan skalabilitasnya (Ramos, 2003). Metode lain yang sempat menjadi perbincangan adalah Word2vec. Word2vec adalah salah satu teknik yang menghasilkan *word embeddings* yang masuk dalam kategori *predictive* model (Mikolov et al., 2013). Teknik Word2vec melakukan *scanning* pada korpus yang besar dengan jumlah jendela yang mirip dengan GloVe, tetapi melakukan pendekatan probabilitas untuk membentuk vektor pada setiap kata dengan membangun neural networks sederhana. Dibandingkan dengan TF-IDF, Word2vec mempunyai keunggulan. Yakni mampu merepresentasikan setiap kata dengan memberikan korelasi makna pada setiap kata. Setelah Word2vec dipublikasikan, *word embeddings* menjadi pembicaraan hangat dikalangan ilmuwan NLP dan tak lama setelahnya bermunculan teknik-teknik optimasi *word embeddings* yang lain. Salah satunya adalah GloVe (Pennington et al., 2014). Pada hasil risetnya, ternyata GloVe mampu mengungguli Word2vec dalam segi akurasi, kecepatan, dan kemampuan dalam beradaptasi ketika sistem akan memperbesar skala. Ini adalah alasan mengapa algoritma GloVe digunakan pada tugas akhir kali ini. Pada tugas akhir ini pula, GloVe diimplementasikan pada korpus berbahasa Indonesia.

Studi literatur tentang CNN berpusat pada publikasi-publikasi ilmiah penerapan arsitektur CNN untuk kasus NLP. Dikarenakan sifat dari CNN yang cenderung eksperimental dalam menentukan desain yang tepat untuk arsitekturnya, maka arsitektur CNN pada implementasi tugas akhir ini akan mulai dibangun dengan cara

mengombinasikan beberapa referensi. Yaitu pada publikasi yang dilakukan oleh Kalchbenner et al. (2014) dengan judul "A Convolutional Neural Network for Modelling Sentences", oleh Hu et al. (2014) dengan judul "Convolutional Neural Network Architectures for Matching Natural Language Senteces", dan oleh Kim (2014) dengan judul "Convolutional Neural Networks for Sentence Classification".

Terakhir adalah studi literatur terhadap teknik *Transfer Learning*. Transfer Learning sendiri masih jarang diaplikasikan pada kasus-kasus NLP sehingga studi lebih lanjut secara mendalam sangat diperlukan pada bagian ini. Setidaknya ada dua publikasi ilmiah yang menjadi acuan utama. Yaitu, publikasi oleh Pan et al. (2010) dengan judul "A Survey on Transfer learning" yang menjadi acuan teori-teori dasar tentang Transfer Learning dan publikasi oleh Yosinski et al. (2014) dengan judul "*How transferable are features on deep learning?*" yang menjadi acuan implementasi transfer learning pada algoritma deep learning.

4. Desain metode

Desain metode dalam tugas akhir ini secara garis besar terbagi menjadi tiga. Pertama, desain program dalam mengubah korpus menjadi *word embeddings* dengan menggunakan algoritma GloVe. Kedua, adalah proses *machine learning* dengan menggunakan CNN hingga didapatkan hasil evaluasi berupa data kuantitatif. Terakhir, adalah desain program untuk melakukan Transfer Learning di mana proses ini melibatkan proses *machine learning* dengan $\mathcal{D}_S$ dan $\mathcal{D}_T$.

5. Implementasi metode

Perangkat keras yang digunakan adalah perangkat keras yang mendukung *graphical processing unit* (GPU) NVIDIA GeForce. Sementara, sistem operasi yang digunakan adalah berbasis Linux demi kemudahan instalasi berbagai macam perangkat lunak yang dibutuhkan. Pada tugas akhir kali ini, perangkat keras dan sistem operasi yang digunakan mempunyai spesifikasi sebagai berikut:

- Intel Core i7-7700 3.6GHz (4 Cores)

- NVIDIA GeForce GTX 1060 3GB
- 8GB of RAM
- Ubuntu 16.04 64-bit

Sedangkan untuk environment perangkat lunak, digunakan bahasa pemrograman Python 2.7 dengan berbagai macam library pendukungnya seperi NumPy, Panda, SciPy, dan seluruh library built-in pada Python 2.7. Semua library tersebut digunakan untuk melakukan manipulasi dan rekayasa data-data mentah sehingga memenuhi kualifikasi untuk menjadi masukan pada library utama.

Pada tugas akhir kali ini, kode GloVe tidak ditulis dari awal. Melainkan menggunakan *source code* GloVe yang dipublikasikan secara terbuka di alamat website pusat penelitan NLP di Stanfor University pada alamat: *https://nlp.stanford.edu/projects/glove*.

Masukan untuk algoritma GloVe adalah hasil dump data artikel-artikel dari Wikipedia Bahasa Indonesia yang dapat diakses melalui alamat: *https://dumps.wikimedia.org/idwiki/latest* dan data yang diambil adalah yang terformat XML. Sebelum diproses oleh algoritma GloVe, terlebih dahulu dilakukan parsing pada data yang baru saja diunduh.

Sebelum melakukan rekayasa dalam membentuk matriks yang dapat menjadi masukan CNN, terlebih dahulu dilakukan berbagai macam langkah *preprocessing* yang melibatkan library Gensim. Instalasi cukup dengan mengetikkan `pip install gensim`.

Lalu, untuk mengimplementasikan algoritma CNN dilakukan terlebih dahulu instalasi TensorFlow untuk GPU yang langkah-langkahnya instalasinya pada komputer dapat diikuti pada alamat: *https://www.tensorflow.org/install*. Sangat dianjurkan untuk menggunakan GPU dikarenakan proses yang jauh lebih cepat dibandingkan dengan menggunakan CPU.

Kemudian, dilakukan instalasi library Keras yang prosesnya dapat diikuti di alamat: *https://keras.io*. Keras adalah library yang dibangun di atas TensorFlow. Keras bertugas mempermudah dalam melakukan pembangunan pada arsitektur Neural Networks sehingga tidak perlu melakukan *coding-from-scratch* pada TensorFlow.

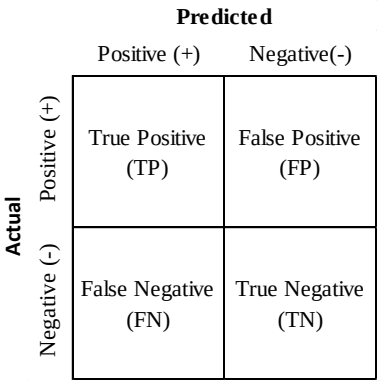
Terakhir, library Scikit-Learn juga digunakan untuk melakukan evaluasi perhitungan akurasi, presisi, *recall*, dan *F-Measure*. Selain itu, Scikit-Learn bisa sangat berguna untuk implementasi berbagai tools pada machine learning saat implementasi tugas akhir ini. Instalasi Scikit-Learn bisa menggunakan pip, conda, atau dengan cara lain yang bisa diikuti pada alamat: *http://scikit-learn.org/stable/install.html*.

6. **Uji Coba dan Evaluasi**

Uji coba pada tugas akhir ini dilakukan untuk mengetahui apakah penggunaan metode ini sudah tepat demi tercapainya tujuan dari tugas akhir ini. Uji coba terlebih dahulu dibuat skenarionya agar pengujiannya tepat sasaran dan sesuai dengan tujuan yang diinginkan.

Evaluasi dalam tugas akhir ini bermanfaat untuk mendapatkan nilai kuantitatif dari skenario uji coba yang dilakukan. Pada tugas akhir ini, metrik yang digunakan adalah *accuracy*, *precision*, *recall*, dan *F-Measure*. Hasil keempat metrik ini diintepretasikan untuk menjawab apakah tujuan dari tugas akhir ini tercapai.

Sebelumnya, perlu diketahui istilah-istilah dalam menentukan rumus-rumus evaluasi menggunakan *confusion matrix* (yang dijelaskan pada Gambar 1.1 yang tediri dari: TP, TN, FP, dan FN.



***Gambar 1.1**. Confusion Matrix.*

Di mana TP adalah *True Prositive* yang artinya adalah jumlah kebenaran antara hasil klasifikasi dengan jumlah seluruh data. TN adalah *True Negative* yang menyatakan jumlah hasil klasifikasi yang diindikasikan benar, tetapi sesungguhnya salah. FP merupakan *False Positive* yakni jumlah dari hasil klasifikasi yang diindikasikan salah, tetapi sesungguhnya benar. Dan FN atau *False Negative* merupakan jumlah dari kesamaan hasil klasifikasi dan yang sesungguhnya adalah salah.

Penjelasan metrik evaluasi *accuracy*, *precision*, *recall*, dan *F-Measure* yang lebih rinci dipaparkan sebagaimana berikut:

a. Accuracy

Akurasi adalah tingkat kedekatan antara nilai prediksi dengan nilai aktualnya. Akurasi dapat dihitung dengan membagi nilai $TP + TN$ dengan jumlah seluruh data sehingga akurasi dapat dirumuskan dengan

$$accuracy = \frac{TP + \sum TN}{TP + \sum TN + \sum FP + \sum FN}. \quad (1.1)$$

Dalam tugas akhir ini, kelas yang digunakan akan lebih dari dua. Maka dari itu, pada Gambar 1.2 bagian (a), diilustrasikan bagaimana mendapatkan akurasi pada perspektif kelas A pada *confusion matrix* tiga kelas.

b. Precision

Precision atau presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Precision* dapat dihitung dengan membagi TP dengan jumlah dari TP dan FP sehingga *precision* dapat dirumuskan menjadi

$$precision = \frac{TP}{\sum TP + \sum FP}. \quad (1.2)$$

Pada perspektif kelas A, nilai *precision* dapat diambil dengan cara melihat ilustrasi *confusion matrix* pada Gambar 1.2 bagian (b).

c. Recall

Recall adalah nilai dari ketepatan informasi yang diprediksi relevan pada suatu kelas terhadap data asli pada kelas tersebut.

Recall dapat dihitung dengan membagi TP dengan jumlah TP dan FN sehingga *recall* dapat dirumuskan dengan

$$recall = \frac{TP}{\sum TP + \sum FN}. \quad (1.3)$$

Pada perspektif kelas A, nilai *recall* dapat diambil dengan melihat ilustrasi *confusion matrix* pada Gambar 1.2 bagian (c).

d. F-Measure

F-Measure memanfaatkan nilai presisi dan *recall*. Memisahkan dokumen-dokumen yang mirip kadang lebih buruk dibandingkan dengan menempatkan pasangan dokumen yang tidak mirip ke dalam satu kelompok. Oleh karena itu, digunakan *F-Measure* dengan nilai FN yang lebih kuat dibandingkan dengan nilai FP. *F-Measure* atau F_1 dapat dihitung dengan menggunakan

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (1.4)$$

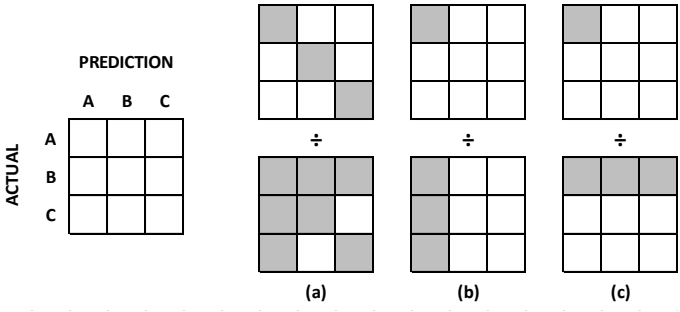
Kemudian, hasil dari masing-masing perspektif pada setiap kelas dirata-rata untuk mendapatkan hasil akhir dari *accuracy*, *precision*, *recall*, dan *F-Measure*.

Evaluasi dilakukan dengan cara membagi data menjadi data latih dan data tes. Pertama, dilakukan prediksi dengan menggunakan model produk dari sistem berdasarkan data tes. Lalu, hasil dari kelas-kelas prediksinya dicocokkan dengan kelas/label pada data tes. Setelah itu, baru dihitung hasil dari *accuracy*, *precision*, *recall*, dan *F-Measure*-nya.

7. Penyusunan buku tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi metode yang telah dibuat. Sistematika penulisab buku tugas akhir ini secara garis besar antara lain:

1. Pendahuluan
    a. Latar Belakang
    b. Rumusan Masalah



Gambar 1.2. *Ilustrasi confusion matrix pada 3 kelas.*

c. Batasan Tugas Akhir
d. Tujuan
e. Manfaat
f. Metodologi

2. Tinjauan Pustaka
3. Desain
4. Implementasi
5. Skenario Uji Coba
6. Uji Coba dan Evaluasi
7. Kesimpulan dan Saran
8. Daftar Pustaka

BAB II
DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar tugas akhir ini.

2.1 Chatbots

Chatbots adalah program komputer yang berinteraksi dengan manusia menggunakan bahasa natural (bahasa sehari-hari yang manusia gunakan). Teknologi ini pertama kali diuji coba pada tahun 1960 yang diberi nama dengan ELIZA (Weizenbaum, 1960). Saat itu, *chatbots* diuji coba untuk mengetahui apakah sistem ini dapat mengelabui manusia layaknya berinteraksi seperti manusia sungguhan.

Zadrozny et al. (2000) sepakat bahwa cara terbaik dalam memfasilitasi interaksi manusia dan komputer adalah memberikan kemungkinan pada pengguna "menyampaikan kepentingannya, keinginannya, atau *query* secara langsung dan natural, dengan bicara, mengetik, atau menunjuk." Ini adalah salah satu alasan utama mengapa pengembangan *chatbots* mulai dilakukan. Karena, *chatbots* merupakan salah satu implementasi yang diutarakan oleh Zadrozny et al. (2000) yang menerapkan konsep *conversational interface*.

Dalam sejarah perkembangannya, *chatbots* tercipta dalam berbagai jenis. ELIZA dan ALICE adalah contoh *chatbots* sebagai media hiburan, Siri atau Google Assitant adalah contoh *chatbots* sebagai mesin pencari (*information retrieval*), atau Happy Assistant System (Chai et al., 2001) adalah contoh *chatbots* yang diciptakan seabagai *sales* dari *e-commerce*. Dapat disimpulkan, *chatbots* telah menjelma menjadi alternatif dari banyak area pekerjaan strategis.

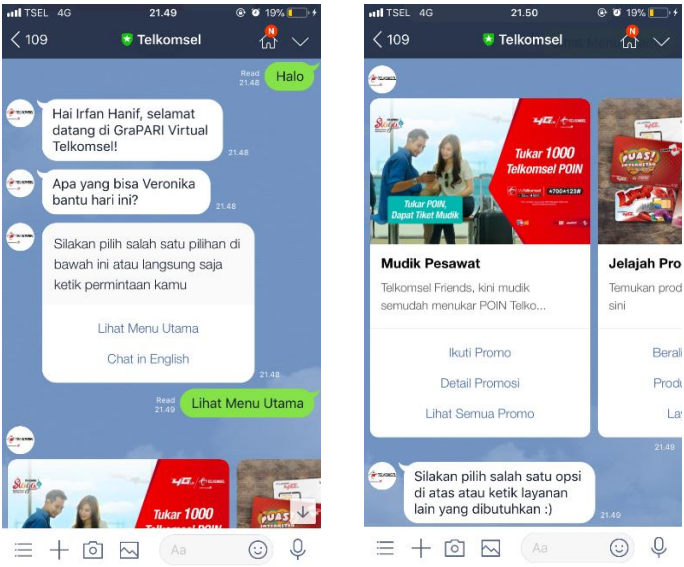
Abu Shawar dan Atwell (2007) menyimpulkan bahwa tujuan dari dibangunnya *chatbots* adalah sebagai alat untuk menolong dan membantu dalam menyelesaikan pekerjaan manusia dengan berinteraksi menggunakan bahasa sehari-hari, tetapi bukan menggantikan manusia itu sendiri. *Chatbots* akan sangat

membantu pada pekerjaan sederhana yang sifatnya repetitif sehingga manusia bisa mengerjakan tugas lain yang sifatnya lebih kompleks. Dan yang paling penting, percakapan yang terbaik tentunya adalah antara manusia dan manusia. *Chatbots* takkan pernah bisa menipu manusia dalam urusan percakapan yang sangat baik.

Chatbots sendiri sudah cukup familiar di Indonesia. Salah satu penerapannya adalah pada *customer service* virtual salah satu perusahaan telekomunikasi milik pemerintah Indonesia yang dinamai Veronika. Seperti yang bisa dilihat pada Gambar 2.1, Veronika memberikan informasi-informasi seputar produk melalui LINE Messenger yang bisa diajak berinteraksi dengan cukup baik.

2.2 Global Vectors for Word Representations (GloVe)

Global Vectors for Word Representations (Glove) adalah algoritma *unsupervised learning* untuk mendapatkan representasi vektor dari kata-kata (*word embeddings*). Proses *training* model



Gambar 2.1. *Veronika, chatbots asal Indonesia*

GloVe dilakukan dengan cara melibatkan seluruh informasi statistik dari korpus dengan membentuk *word-word co-occurrence matrix.* Proses *training* GloVe lebih efisien dibandingkan dengan teknik lain karena GloVe hanya melibatkan *training* pada elemen matriks yang nilainya bukan nol.

GloVe menghasilkan Word Embeddings yang sangat baik. Hal ini terbukti dengan tingkat keberhasilan hingga 75% pada tes analogi kata. Word Embeddings GloVe juga mengungguli model Word Embeddings yang lain pada tes kemiripan antar kata dan pengenalan nama benda (Pennington et al., 2014). Word Embeddings yang dihasilkan GloVe diproyeksikan berkontribusi besar untuk menangani keragaman *thesaurus.* Hal ini diakibatkan representasi vektor dari Word Embeddings memberikan hasil yang baik pada uji coba *nearest* beigbors. Santika et al. (2015) dalam penelitiannya mengatakan kemampuan model mengenali *thesaurus* dari banyak diksi akan meningkatkan performa klasifikasi teks.

Pada penelitian-penelitian sebelumnya, teknik dalam mendapatkan Word Embeddings terbagi menjadi dua. Yaitu, Metode faktorisasi matriks dan metode berbasis *context-window*. Metode faktorisasi matriks mendapatkan Word Embeddings dengan cara membuat matriks berdasarkan kemunculan kata pada korpus lalu mengonversinya menjadi vektor berdimensi tertentu. Metode yang berbasis *context-window* mendapatkan makna sebuah kata dengan cara mempelajari kata yang muncul di sekitarnya (yang masuk dalam rentang panjang *window*). Metode faktorisasi matriks mempunyai keunggulan dibandingkan metode *context-window* dalam mendapatkan informasi statistik yang merepresentasikan seluruh korpus, karena metode *context-window* hanya melakukan pemindaian pada *context-window* secara lokal saja. Padahal, nilai statistik pengulangan kata pada korpus adalah informasi yang sangat penting. Di sisi lain, mempelajari makna kata dari kata-kata yang ada di sekitarnya adalah ide yang bagus.

Menurut Pennington et al. (2014) metode faktorisasi matriks seperti *Latent Semantic Analysis* (Landauer et al., 1998),

Hyperspace Analogue to Language (Lund dan Burgess, 1996), atau *Positive Pointwise Mutual Information* (Bullinaria dan Levy, 2014) masih memiliki beberapa kekuragan dalam mengonversi informasi statistik menjadi Word Embeddings sehingga ada potensi untuk dikembangkan. Maka, GloVe berusaha menggabungkan kedua metode ini dengan cara melakukan faktorisasi matriks dari korpus yang isinya adalah nilai dari perhitungan kemunculan kata pada *context-window*.

Jika X adalah sebuah *co-occurrence matrix*, maka X_{ij} adalah jumlah kemunculan kata j pada konteks kata i. Kata konteks yang dimaksud di sini adalah semua kata yang masuk dalam rentang *window*. Jika s adalah panjang *window* dan w_j adalah sebuah kata ke-j pada korpus, maka kata konteks adalah setiap kata yang masuk pada rentang $[w_{j-s}, w_{j-1})$ dan $(w_{j+1}, w_{j+s}]$. Matriks X hanya dibangun sebanyak satu kali sehingga perhitungan selanjutnya akan berdasarkan data dari matriks ini. Jika $X_i = \sum_k X_{ik}$ adalah jumlah kemunculan semua kata pada konteks kata i, maka $P_{ij} = P(j|i) = X_{ij}/X_i$ adalah kemungkinan kata j muncul pada konteks kata i.

Setiap kata konteks yang muncul pada kata utama tidak selalu bernilai satu. Jika panjang *window* adalah s, maka $\mu = \{1,2, \dots, s\}$ adalah jarak kata konteks dengan kata utama sepanjang *window* s. Lalu, X_{ij} adalah jumlah kemunculan kata konteks j pada kata utama i sehingga nilai X_{ij} pada waktu t berikutnya $(t+1)$ ditentukan dengan persamaan

$$X_{ij,t+1} = X_{ij,t} + \frac{1}{\mu};\ \ \mu = \{1,2, \dots, s\}\,. \qquad (2.1)$$

Nilai s ini juga sudah ditentukan sebesar 10 karena telah terbukti memberikan hasil yang terbaik (Gambar 4.1). Hal ini didasari

Tabel 2.1. *Contoh ratio kemungkinan kemunculan kata.*

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1,9 x 10^{-4}	6,6 x 10^{-5}	3,0 x 10^{-3}	1,7 x 10^{-5}
$P(k\|steam)$	2,2 x 10^{-5}	7,8 x 10^{-4}	2,2 x 10^{-3}	1,8 x 10^{-5}
$P(k\|ice)/P(k\|steam)$	8,9	8,5 x 10^{-2}	1,36	0,96

dengan asumsi bahwa semakin jauh sebuah kata konteks dengan kata utama, maka semakin sedikit pula informasi keterhubungan antar kata.

Permulaan terbaik dalam mendapatkan vektor kata adalah dengan mempelajari rasio dari kemungkinan kemunculan sebuah kata. Sebagai contoh, dalam bagian korpus yang membahas tentang termodinamika (misal $i = ice$ dan $j = steam$), setiap kata k yang memiliki hubungan yang kuat dengan *ice*, tetapi tidak dengan *steam*, misal $k = solid$, maka rasio dari P_{ik}/P_{jk} harusnya bernilai besar. Sebaliknya, jika kata k memiliki hubungan yang kuat dengan *steam* tetapi tidak dengan *ice*, maka P_{ik}/P_{jk} harusnya bernilai kecil. Jika kata k adalah kata yang memiliki hubungan yang sama-sama kuat atau sama-sama tidak kuat dengan *ice* dan *steam*, maka nilai rasionya mendekati satu.

Pada Tabel 2.1 dapat dilihat hasil dari P_{ik}/P_{jk} dengan $k = solid$ nilai perbandingannya sangat besar karena kata $solid$ lebih dekat dengan kata ice. Hal ini juga terjadi dengan $k = gas$. Nilai perbandingan P_{ik}/P_{jk} menjadi sangat kecil karena kata gas lebih dekat dengan kata $steam$. Jika diambil kata yang tidak lebih dekat dengan ice atau $steam$, misalnya seperti $water$ atau $fashion$, nilai perbandingan ini mendekati angka satu.

Maka dari itu, bisa disimpulkan model umum dirumuskan dalam fungsi F yang terdiri dari variabel w_i, w_j, dan $\widetilde{w}_k$ yang nilainya harus mendekali nilai rasio P_{ik}/P_{jk}. Jika $w \in \mathbb{R}^d$ adalah vektor kata dan $\widetilde{w} \in \mathbb{R}^d$ adalah vektor kata konteks, maka fungsi F dapat didefinisikan menjadi

$$F(w_i, w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}}. \qquad (2.2)$$

Nilai bagian kanan pada Persamaan (2.2) dapat diambil dari matriks X sementara F dapat ditentukan dengan cara mengubah bentuk rasio P_{ik}/P_{jk} dalam bentuk vektor. Karena bentuk vektor adalah linear, maka cara termudah adalah dengan cara mengambil jarak antar vektor. Dengan begitu, fungsi F dapat diadaptasi menjadi

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}. \quad (2.3)$$

Dapat diketahui bahwa sisi kiri dari Persamaan (2.3) adalah vektor dan sisi kanannya adalah skalar. Maka dari itu, vektor dapat diubah menjadi skalar dengan operasi *dot product* sehingga menjadi

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}. \quad (2.4)$$

Bentuk persamaan ini dapat dikembalikan menjadi simetri dengan dua langkah. Pertama, F harus homomorfis antara ($\mathbb{R}$,+) dan ($\mathbb{R}_{>0}$,x) menjadi

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}, \quad (2.5)$$

dengan begitu, Persamaan (2.5) dapat diperbaiki menjadi

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}, \quad (2.6)$$

sehingga Persamaan (2.6) menjadi

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i). \quad (2.7)$$

Selanjutnya, karena $\log(X_i)$ tidak bergantung pada k, maka dapat diubah bentuknya menjadi bias b_i untuk w_i. Akhirnya, persamaan ini dapat kembali simteris dengan menambahkan $\tilde{b}_k$ untuk $\tilde{w}_k$ sehingga persamaannya menjadi

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}). \quad (2.8)$$

Persamaan (2.8) adalah bentuk sederhana yang bagus dari Persamaan (2.2). Namun, terdapat kecacatan pada persamaan ini ketika elemen pada matriks X bernilai nol, padahal nilai nol pada matriks X bisa berjumlah 75-95% dari total elemen pada matriks (tergantung dari besar korpus). Maka dari itu, elemen bernilai nol tidak diproses. Lalu, bagaimana cara menangani kata-kata yang lebih banyak mengandung elemen bernilai nol? Bagaimana memproses kata-kata ini sehingga didapatkan representasi vektor

yang bagus? Pennington et al. (2014) mengusulkan teknik pembobotan baru untuk menangani masalah ini yaitu dengan memberikan fungsi pembobotan $f(X_{ij})$ pada *cost function* sehingga didapatkan model *cost function*

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 , \qquad (2.9)$$

dengan $f(x)$ sebagai fungsi pembobotan yang dirumuskan dengan

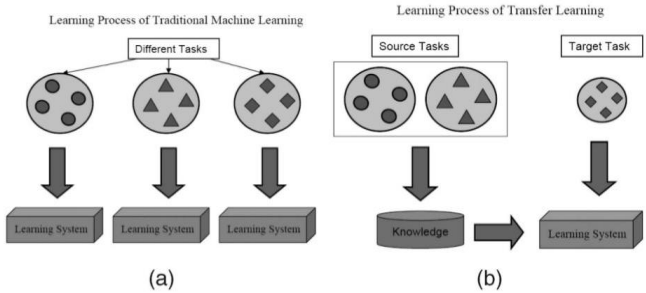
$$f(x) = \begin{cases} (x/x_{\max})^{\alpha}, & x < x_{max} \\ 1, & otherwise \end{cases} . \qquad (2.10)$$

Pennington et al. (2014) dalam publikasinya menetapkan nilai $\alpha = 3/4$ dan $x_{max} = 100$ untuk seluruh eksperimennya. Nilai x_{max} ditentukan sebanyak 100 karena setiap pasangan kata utama dan kata konteks yang muncul 100 kali dianggap terlalu sering muncul sehingga langsung diberi bobot satu.

Pennington et al. (2014) dalam publikasinya memilih algoritma AdaGrad untuk *training*. AdaGrad dipilih karena kemampuan *learning rate*-nya yang beradaptasi pada parameter tertentu. Parameter yang muncul lebih jarang akan mendapatkan update yang lebih besar dibandingkan yang sering muncul. Jadi, AdaGrad sangat cocok untuk matriks GloVe yang *sparse*.

Setelah nilai *cost* dihitung, sebagai bagian penting dari *training* dengan AdaGrad, kita perlu untuk menghitung gradien dari masing-masing parameter. Ditinjau dari fungsi *cost* J pada Persamaan (2.9), gradien dari masing-masing parameter pada kata utama i dan kata konteks j dapat diambil dari turunan fungsi *cost* yang tediri dari

$$\begin{aligned} \nabla_{\tilde{w}_j} J &= 2 \cdot w_i \cdot f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij}\right), \\ \nabla_{w_i} J &= 2 \cdot \tilde{w}_j \cdot f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij}\right), \\ \frac{\partial J}{\partial b_i} &= 2 \cdot f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij}\right), \\ \frac{\partial J}{\partial b_j} &= 2 \cdot f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij}\right). \end{aligned} \qquad (2.11)$$



Gambar 2.2. *Perbedaan proses learning pada (a) traditional machine learning dan (b) transfer learning (Pan et al., 2010).*

Setelah itu, dilakukan update pada setiap parameter dan menyimpan nilai jumlah kuadrat gradien yang sudah dihitung ketika *training* dengan AdaGrad.

Hasil dari *training* menghasilkan dua vektor kata, yaitu W dan $\widetilde{W}$. Dua vektor ini sebenarnya bisa dipilih salah satunya untuk dijadikan *word embeddings* yang merepresentasikan setiap kata. Ternyata, menjumlahkan kedua vektor $W + \widetilde{W}$ memberikan hasil yang terbaik pada tes analogi kata (Pennington et al., 2014).

2.3 Transfer Learning

Teknologi *data mining* dan *machine learning* telah mencapai sukses pada banyak hal termasuk teknik klasfikasi, *clustering*, dan regresi. Tetapi, banyak metode *machine learning* hanya bekerja dengan baik pada dataset yang mempunyai fitur dan distribusi data yang sama. Ketika distribusi data berubah, model pada *machine learning* harus benar-benar dibangun ulang dengan menggunakan data baru yang dikumpulkan lagi (Pan et al., 2010). Kenyataannya, hal ini sangat merugikan sebab diharuskan untuk mengeluarkan banyak usaha dan biaya untuk mengumpulkan data yang sesuai dan membangun ulang model dari awal. Tentunya akan lebih efisien ketika model yang pernah dibangun bisa digunakan lagi untuk masalah yang mirip. Di sinilah metode *Knowledge Transfer* atau *Transfer Learning* dibutuhkan. Seperti yang dijelaskan pada

Tabel 2.2. *Istilah yang digunakan pada Transfer Learning.*

No	Istilah	Penjelasan	Notasi
1	Source Domain	Seluruh hal yang mencakup Source Task.	-
2	Target Domain	Seluruh hal yang mencakup Target Task.	-
3	Source Task	Aktivitas *learning* yang hasil pemodelannya akan digunakan kembali pada Target Task.	$\mathcal{T}_S$
4	Target Task	Aktivitas *learning* yang menjadi tujuan utama penelitian. Pada tahap ini, proses *learning* menggunakan kembali pemodelan hasil dari Source Task.	$\mathcal{T}_\mathcal{T}$
5	Source Dataset	Data yang digunakan pada Source Task.	$\mathcal{D}_S$
6	Target Dataset	Data yang digunakan pada Target Task.	$\mathcal{D}_\mathcal{T}$

Gambar 2.2, proses Trasfer Learning akan memberikan alur yang lebih efisien karena model yang pernah digunakan sebelumnya dapat digunakan lagi untuk sistem yang lain.

Sebelum masuk lebih jauh dalam pembahasan Transfer Learning, ada beberapa istilah yang digunakan dalam proses Transfer Learning yang dijelaskan pada terlebih dahulu. Istilah beserta notasi ini akan digunakan untuk mempermudah penjelasan dalam beberapa eksperimen tentang Transfer Learning. Termasuk penjelasan saat implementasi maupun skenario uji coba yang dilakukan pada tugas akhir ini. Penjelasan ini akan diringkas pada Tabel 2.2.

Berdasarkan definisi dari Transfer Learning, terdapat 3 kategori dari Transfer Learning (Pan et al., 2010). Untuk lebih jelasnya, perbedaan masing-masing teknik dijelaskan pada Tabel 2.3. Secara definisi, ketiga teknik itu adalah:

1. *Inductive Transfer Learning*. Pada i*nductive transfer learning*, Target Task berbeda dengan Source Task, tidak peduli apakah $\mathcal{D}_S$ sama dengan $\mathcal{D}_\mathcal{T}$.
2. *Transductive Transfer Learning*. Pada teknik ini, Target Task dan Source Task sama ketika $\mathcal{D}_S$ dan $\mathcal{D}_\mathcal{T}$ berbeda.

Tabel 2 3. *Perbedaan teknik pada Transfer Learning.*

Settings	Related Access	$\mathcal{D}_S$ Labels	$\mathcal{D}_T$ Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught learning	Unavailable	Available	Regression, Classification
Transductive Transfer Learning	Domain adaptation, Sample Selection Bias, Co-variate shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Regression, Classification

3. *Unsupervised Transfer Learning*. Teknik ini mirip dengan *inductive transfer learning*. Target Task dan Soure Task berbeda namun mempunyai hubungan dengan Target Task. *Unsupervised transfer learning* bertugas untuk fokus pada data-data yang tidak mempunyai label seperti *clustering*, reduksi dimensi, atau *density estimation*.

Transfer Learning pernah dipelajari oleh Yosinski et al. (2014) dan diimplementasikan pada algoritma *deep learning*. Terdapat satu fenomena unik yang ditemukan: ketika data citra dilatih, hasil dari layer pertama selalu mirip dengan *Gabor filters* atau *color blobs*. Fenomena ini tidak hanya terjadi pada satu dataset tetapi terjadi pada setiap dataset dengan tujuan berbeda seperti *supervised image classification* (Krizhevsky, 2012), *unsupervised density learning* (Lee, 2009), dan *unsupervised learning of sparse representation* (Le, 2011).

Hasil penelitian yang dilakukan Yosinski et al. (2014) menunjukkan bahwa layer teratas dari *deep learning* bisa diasumsikan mengerjakan tugas yang umum. Sedangkan layer yang lebih dalam mengerjakan tugas yang semakin spesifik. Perbedaan dari tugas model pada Target Domain dan Source

Domain juga mempengaruhi transfer *knowledge*. Semakin jauh bedanya, semakin sulit untuk dilakukan transfer *knowledge*. Walaupun begitu, implementasi transfer *knowledge* dengan tujuan Yang jauh berbeda tetap lebih baik dibandingkan dengan melakukan inisialisasi nilai bobot secara acak. Kesimpulan dari penelitian dari Yosinski et al. (2014) juga mengatakan bahwa transfer *knowledge* (proses Transfer Learning) akan memberikan performa yang lebih baik pada *deep learning* setelah *tuning* (mendapatkan nilai variabel-variabel pada Neural Network) yang cocok.

2.4 Convolutional Neural Networks (CNN)

Awalnya Convolutional Neural Networks (CNN) ditemukan untuk kasus *machine learning* pada citra. Namun, CNN ternyata memberikan hasil yang sangat bagus pada kasus-kasus *Natural Language Processing* (NLP) seperti *semantic parsing* (Yih et al., 2014), *search query retrieval* (Shen et al., 2014), *sentence modelling* (Kalchbrenner et al., 2014), dan kasus NLP lainnya (Collobert et al., 2011).

Pada penelitiannya yang berjudul "*Convolutional Neural Networks for Sentence Classification*", Kim (2014) mengombinasikan CNN sederhana dengan Word Embeddings Word2Vec yang dikembangkan oleh Mikolov et al. (2013). Hasilnya, CNN yang hanya menggunakan satu layer konvolusi ini memberikan hasil yang sangat baik pada kasus klasifikasi kalimat. Dalam perbandingannya dengan metode lain yang termutakhir, metode yang dikembangkan Kim (2014) berhasil mengungguli 4 dari 7 kasus yang pernah dipublikasi.

CNN pada kasus teks bekerja sama persis dengan CNN pada kasus citra. Perbedaannya adalah layer konvolusi dan layer-layer lain yang mengikutinya berukuran satu dimensi. Sedangkan pada citra, layer-layernya bisa berukuran dua atau tiga dimensi.

Layer konvolusi bekerja dengan cara melakukan pemindaian pada setiap elemen input (dalam hal ini adalah vektor satu dimensi yang merepresentasikan sebuah kalimat/dokumen). Layer ini bertugas untuk menangkap fitur yang signifikan terhadap proses

klasifikasi yang informasinya didapatkan pada layer-layer klasifikasi. Karena bentuk masukan pada CNN ini adalah Word Embeddings, maka menyerahkan kepada layer konvolusi untuk menentukan fitur yang lebih signifikan adalah salah satu cara yang terbaik.

Jika dilihat pada Gambar 2.3, bagian yang direpresentasikan garis putus-putus adalah *strides*. Bagian yang berwarna biru adalah matriks input, bagian yang berwarna abu-abu adalah *filter* atau *kernel*, bagian berwarna hijau adalah matriks yang menyimpan hasil dari perhitungan *dot product* atau disebut dengan output. Berdasarkan gambar di atas, *strides*-nya berukuran 2x2, *kernel*-nya berukuran 4x4, inputnya berukuran 5x5, dan outputnya berukuran 6x6. Dalam implementasi untuk data teks, maka bisa diadaptasi sebagai berikut: *strides* berukuran 2, *kernel* berukuran 4, input berukuran 5, dan output berukuran 6 akibat data teks yang satu dimensi. Begitulah kurang lebih anatomi pada layer konvolusi.

Dalam kasus klasifikasi teks dengan Word Embeddings, pada tugas akhir ini, vektor yang membentuk kalimat (vektor dokumen) disusun dengan cara merata-rata setiap vektor kata yang menyusun kalimat. Jika s_i adalah sebuah kalimat ke-i, maka s_i terdiri dari beberapa kata w sehingga $s_i = \{w_1, w_2, \dots, w_n\}$. Pertama, untuk mendapatkan penjumlahan vektor dokumen $\vec{s}_i$, lakukan penjumlahan setiap vektor kata $\vec{w}_i$ pada dokumen s_i sehingga

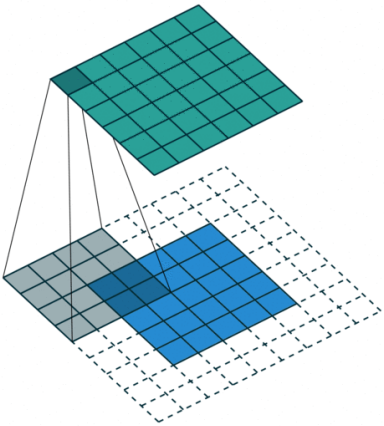
$$\vec{s}_i = \sum_{i=1}^{n} \vec{w}_i \,. \tag{2.12}$$

Lalu, untuk mendapatkan vektor dokumen $\vec{d}(s_i)$, $\vec{s}_i$ dibagi dengan normalisasi dari $\vec{s}_i$ sehingga persamaannya dirumuskan menjadi

$$\vec{d}(s_i) = \frac{\vec{s}_i}{norm(\vec{s}_i)}, \tag{2.13}$$

dan $norm(\vec{s}_i)$ bisa didapatkan nilainya dengan persamaan

$$norm(X) = \sqrt{\sum_{j=1}^{n} |X_j|^2} \,. \tag{2.14}$$



Gambar 2.3. *Cara kerja sebuah Convolution Layer.*

Kenter et al. (2016) mengatakan bahwa hasil dari merata-rata Word Embeddings telah terbukti memberikan hasil yang efektif dalam merepresentasikan Sentence Embeddings atau vektor dokumen. Dengan memanfaatkan layer konvolusi pada CNN, CNN akan mengekstrasi fitur penting pada vektor dokumen sehingga hasil klasifikasi lebih akurat.

Model CNN yang digunakan pada tugas akhir ini kurang lebih terbagi menjadi dua. Yang pertama adalah bagian konvolusi dan yang kedua adalah bagian klasifikasi. Bagian klasifikasi terdiri dari *hidden layer* dan *output layer* yang diisi oleh neuron-neuron. Pada setiap elemen output pada layer konvolusi terhubung dengan setiap layer pada *hidden layer*. Begitu juga dengan setiap neuron-neuron pada *hidden layer* terhubung dengan neuron-neuron pada *output layer*.

Layer konvolusi dan *hidden layer* akan menggunakan fungsi aktivasi Rectified Linear Units (ReLU). Krizhevsky et al. (2012) dalam publikasinya mengatakan bahwa ReLU meningkatkan

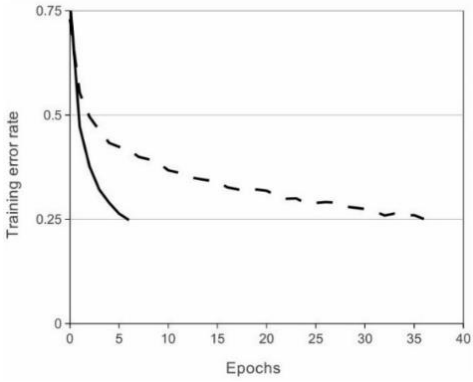
performa *training* 6x lebih cepat dibandingkan fungsi Tanh. Gambar 2.4 menunjukan ReLU menggapai *training error* pada level 25% dengan jauh lebih cepat dibandingkan Tanh pada dataset CIFAR-10. Persamaan (2.15) menunjukan persamaan ReLU Nonlinearity di mana jika x lebih kecil dari 0, maka nilai $\phi(x)$ adalah 0 dan jika lebih besar dari 0, maka nilai $\phi(x)$ adalah nilai x itu sendiri. Persamaan ReLU dirumuskan menjadi

$$\phi(x) = \max(0, x). \tag{2.15}$$

Pada layer output, fungsi aktivasi yang digunakan adalah Softmax. Softmax berfungsi untuk mendistribusikan hasil *learning* atau prediksi dari layer-layer sebelumnya pada rentang [0,1] dengan hasil akhir total seluruh nilai elemennya adalah 1. Nilai tertinggi yang dihasilkan Softmax adalah kelas hasil prediksi dari CNN. Dengan j adalah index dari setiap elemen dari z dan K adalah jumlah seluruh elemen z, persamaan Softmax dijabarkan menjadi

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad j = \{1,2,\dots,K\}. \tag{2.16}$$

Untuk menghindari *overfitting*, CNN dilengkapi dengan regularisasi Dropout. Dropout akan memberikan nilai 0 pada



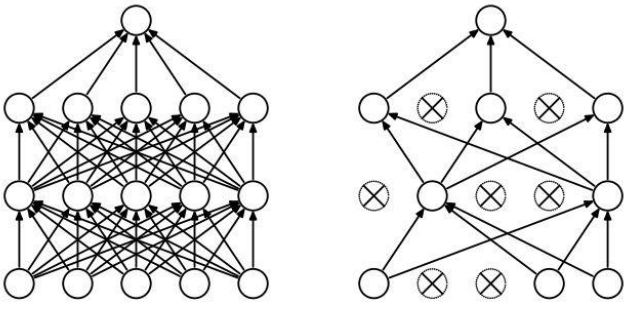
Gambar 2.4. *ReLU dan Tanh pada CIFAR-10 (Kryzhevsky et al., 2012).*

sebuah *neuron* secara acak ketika proses *backpropagation* dan *forward learning*. Ini adalah teknik simpel agar Neural Network tidak terjebak dalam *overfitting* (Srivastava et al., 2014).

Jika ditentukan regularisasi nilai Dropout adalah p, maka sebuah neuron akan di-"dropout" dengan mengganti nilai prediksinya menjadi 0 ketika hasil prediksinya bernilai p. Jadi, setiap proses *learning* akan menggunakan arsitektur CNN yang berbeda-beda sehingga menghasilkan model yang lebih *robust*. Pada saat proses *testing*, setiap prediksi dari neuron yang layernya diregularisasi dengan Dropout akan dikali dengan p sehingga hasilnya sama dengan saat proses *training*. Ilustrasi dari Dropout dijelaskan pada Gambar 2.5.

2.5 Adaptive Subgradient Methods (AdaGrad)

AdaGrad adalah algoritma optimisasi berbasis gradien yang memungkinkan *learning rate* beradaptasi berdasarkan parameternya (Duchi et al., 2011). Pennington et al. (2014) menggunaka AdaGrad untuk melakukan *training* pada GloVe. Hal ini dikarenakan dalam membangun *word-word co-occurrence matrix*, matriks yang dihasilkan ternyata sangat *sparse* (75-95% mengandung elemen bernilai nol) sehingga dibutuhkan algoritma



Gambar 2.5. *Perbedaan arsitektur Neural Network dengan Dropout (Srivastava et al., 2014).*

yang mampu beradaptasi pada keragaman jumlah kemunculan kata.

Ketika *learning rate* diinisiasi denga nilai yang kecil, hal ini akan bermasalah pada kata yang jarang muncul karena kontribusinya untuk meminimalisasi *cost function* menjadi sangat jarang. Sedangkan jika *learning rate* diinisiasi terlalu besar, proses *learning* pada kata yang sering muncul akan melompat terlalu jauh sehingga tidak akan sampai pada titik optimum. AdaGrad digunakan untuk memecahkan masalah ini.

Pada algoritma Stochastic Gradient Descent, biasa disingkat SGD (Kiefer dan Wolfowitz, 1952), setiap parameter θ_i diupdate menggunakan nilai *learning rate* η yang sama. AdaGrad menggunakan pendekatan yang lain. AdaGrad menggunakan nilai η yang berbeda pada setiap parameter θ_i pada setiap waktu t. Jika $g_{t,i}$ adalah gradien dari fungsi objektif parameter θ_i pada waktu t, maka nilai dari $g_{t,i}$ dapat didapatkan dari

$$g_{t,i} = \nabla_\theta J(\theta_{t,i}) . \tag{2.17}$$

Pada SGD, update pada setiap parameter θ_i adalah

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} , \tag{2.18}$$

sedangkan AdaGrad memodifikasi nilai η pada setiap waktu t untuk setiap parameter θ_i berdasarkan nilai gradien sebelumnya hasil perhitungan dari gradien θ_i. Jika $G_{t,i}$ adalah jumlah seluruh kuadrat dari gradien-gradien hasil perhintungan sebelumnya pada parameter θ_i dan waktu t, maka update parameter pada AdaGrad dapat dijelaskan menjadi

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} \cdot g_{t,i} . \tag{2.19}$$

Sedangkan ϵ adalah sebuah nilai desimal untuk menghindari pembagian dengan nol. Nilai ϵ biasanya bernilai $1e - 8$.

2.6 Adaptive Moment Estimation (Adam)

Adaptive Moment Estimation, atau biasa yang disingkat Adam (Kingma dan Lei Ba, 2015), adalah metode lain dari *gradient descent* pengembangan dari AdaGrad yang menerapkan *learning*

rate adaptif pada setiap parameter tertentu. Adam menganalogikan algoritmanya sebagai sebuah bola menggelinding yang mempunyai gaya gesek dan momentum dalam proses mencari titik optimum. Sama seperti AdaGrad, Adam juga menyimpan jumlah kuadrat dari nilai gradien hasil perhitungan sebelumnya. Sebuah parameter θ_i dalam Adam diupdate dengan persamaan

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \cdot \hat{m}_t \,, \tag{2.20}$$

dengan $\hat{m}_t$ dan $\hat{v}_t$ didapatkan dari persamaan

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{2.21}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{2.22}$$

dan m_t dan v_t didapatkan dengan persamaan

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{2.23}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2. \tag{2.24}$$

Persamaan (2.21) dan (2.22) dibuat karena Kingma dan Lei Ba (2015) menemukan bahwa nilai m_t dan v_t terlalu bias dengan angka nol. Khususnya, pada saat langkah awal dan juga karena nilai β_1 dan $\beta_2$terlalu dekat dengan angka 1.

Kingma dan Lei Ba (2015) mengusulkan nilai tetap $\beta_1 = 0.9$ dan nilai $\beta_1 = 0.999$. Sedangkan nilai $\varepsilon = 10^{-8}$ digunakan untuk mencegah pembagian dengan nol. Metode Adam telah terbukti bekerja lebih baik dibandingkan dengan metode *adaptive learning* lain seperti AdaGrad, AdaDelta, dan RMSProp sehingga membuatnya menjadi algortima yang kini sering digunakan untuk melatih Neural Networks.

2.7 Cosine Similarity

Cosine Similarity adalah teknik mengukur kemiripan antara dua vektor pada ruang spasial. Teknik ini mengukur kemiripan dengan cara meninjau dari orientasi sudut antara dua vektor dan mengonversinya menjadi nilai *cosinus*. Pada vektor yang

mempunyai orientasi yang sama, maka sudut yang diciptakan kedua vektor adalah 0 derajat sehingga nilai *cosinus* yang dihasilkan adalah 1. Sebaliknya, pada vektor yang memiliki orientasi saling tegak lurus, maka sudut yang diciptakan kedua vektor adalah 90 derajat sehingga nilai *cosinus* yang dihasilkan adalah 0.

Persamaan Cosine Similarity adalah sebagai berikut. Jika X dan Y adalah kedua vektor yang ingin diukur kemiripannya; X_i dan Y_i adalah nilai elemen vektor X dan Y pada indeks ke-i; dan n panjang vektor X dan Y, maka persamaan Cosine Similarity didefinisikan menjadi

$$cos(\theta) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}}. \qquad (5.1)$$

BAB III
DESAIN

Pada bab ini akan dijabarkan desain tugas akhir yang meliputi tahap-tahap penyelesaian tugas akhir baik secara umum, maupun algoritma-algoritma yang mendukungnya secara khusus.

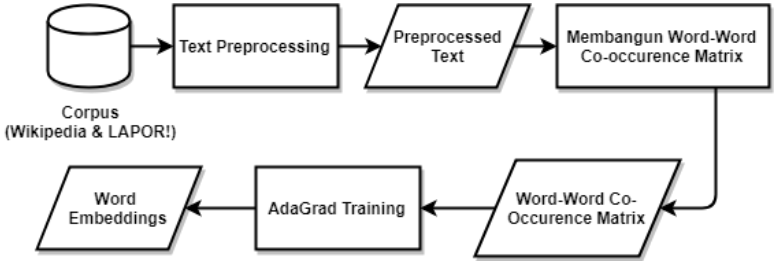
3.1 Desain Aplikasi GloVe

Output dari GloVe adalah vektor kata yang mereprentasikan makna dari setiap kata (*word embeddings*). Untuk mendapatkan Word Embeddings, GloVe membutuhkan korpus yang besar untuk jadi masukannya. Yang dimaksud dengan besar di sini adalah korpus yang mempunyai jumlah token yang sangat banyak jumlahnya (bisa puluhan juta atau bahkan mencapai milyar). Jika korpus yang digunakan tidak cukup besar, maka GloVe akan gagal memberikan representasi makna pada setiap kata.

Seperti yang terlihat pada Gambar 3.1, langkah-langkah dalam mendapatkan Word Embeddings GloVe adalah sebagai berikut.

3.1.1. *Text Preprocessing*

Pertama, dilakukan *parsing* dari format XML untuk mendapatkan konten dari setiap tulisan di Wikipedia. Untuk mendapatkan Korpus Wikipedia Bahasa Indonesia berformat XML, berkasnya dapat diunduh secara terbuka di alamat https://dumps.wikimedia.org/idwiki/latest. Selama proses *parsing* ini, pada setiap artikel di Wikipedia dilakukan *tokenization*; *tokenization* adalah proses memisahkan setiap kata pada korpus



***Gambar 3.1**. Desain aplikasi GloVe.*

	saya	neural	network	sangat	menarik	senang	belajar
saya	0	0	0	1	1	1	1
neural	0	0	2	1	0	1	1
network	0	2	0	1	1	0	1
sangat	0	1	1	0	1	0	0
menarik	1	0	1	1	0	1	0
senang	1	1	0	0	0	0	1
belajar	1	1	1	0	0	1	0

Gambar 3.2. *Contoh matriks word-word co-occurrence.*

menjadi token-token. Lalu setiap token ini dijadikan huruf kecil atau *lower case*. Setiap token ini kemudian dilakukan proses *regular expression* untuk menghilang seluruh karakter selain huruf (baik angka, maupun karakter-karakter lain). Hasil token-token yang telah diubah huruf kecil ini kemudian diletakkan ke dalam sebuah *file* dengan memisahkan setiap tokennya dengan spasi. Korpus berikutnya adalah korpus dokumen data "LAPOR!". LAPOR! adalah portal daring penyedia layanan laporan masyarakat untuk mengadukan kinerja pemerintah secara nasional. Dataset ini bisa diunduh secara terbuka pada alamat https://data.go.id (portal pemerintah yang menyimpan *open data* nasional). Dokumen-dokumen pada dataset ini juga dijadikan korpus. Prosesnya adalah mengubah bentuk setiap dokumen menjadi token-token *lower case*. Proses *text preprocessing* lain seperti mengubah setiap kata menjadi kata dasar (*stemming*), membuang kata yang sangat sering muncul (*stopwords*), atau menentukan unsur dari setiap kata (*part of speech tagging)* tidak dilakukan.

3.1.2. Membangun *Word-Word Co-Occurrence Matrix*

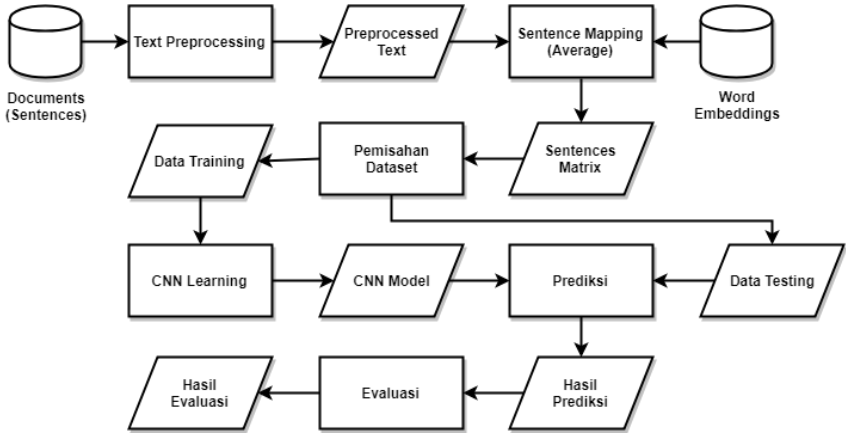
Berikutnya, sebelum melatih Word Embeddings menggunakan AdaGrad, perlu dibangun sebuah matriks *word-word co-occurrence*. Matriks *word-word co-occurrence* sebenarnya adalah tabel dua dimensi yang kolom-kolomnyanya adalah kata konteks dan baris-barisnya adalah kata target. Proses membentuk matriks *word-word co-occurrence* dilakukan dengan cara memindai seluruh token yang ada pada korpus. Sebagai contoh cara membangun sebuah matriks *word-word co-occurrence*, ambil contoh sepenggal korpus seperti berikut ini.

```
neural network sangat menarik
saya sangat senang belajar neural network
```

Setelah menentukan *window* dengan panjang tertentu (misal ditentukan panjang *window* adalah 2), setiap elemen pada matriks bisa diisi nilainya. Pemindaian akan dimulai pada kata “neural” sebagai kata target dengan kata konteks “network” dan “sangat” sehingga elemen dengan baris kata “neural” dan kata konteks “network” dan “sangat” nilainya masing-masing ditambah 1. Lalu kata target akan bergeser pada kata “network” sehingga kata konteksnya adalah “neural”, “sangat”, dan “menarik”. Lalu kata target bergeser lagi menjadi kata “sangat” dengan kata konteks “neural”, “network”, “menarik”, dan “saya”. Dengan mengabaikan pembobotan jarak antara kata target dengan kata konteks, matriks *word-word co-occurrence* akan menjadi seperti pada Gambar 3.2.

3.1.3. *AdaGrad Training*

Setelah matriks *word-word co-occurrence* dibangun, parameter Word Embeddings dengan panjang tertentu akan diinisialisasi secara random. Baik Word Embeddings pada kata target, maupun Word Embeddings pada kata konteks. Begitu pula dengan nilai bias pada masing-masing Word Embeddings. Proses melatih dengan AdaGrad melibatkan *cost function* seperti yang tertera pada Persamaan 2.9.



Gambar 3.3. *Desain aplikasi CNN.*

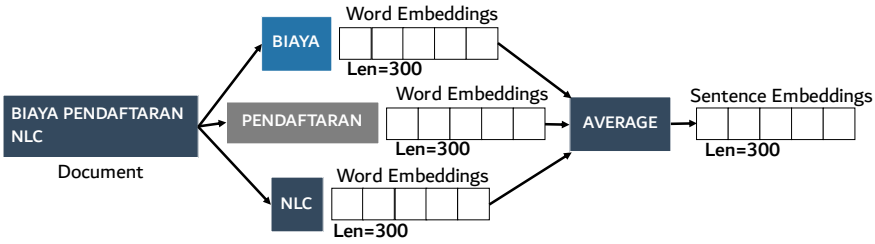
Proses melatih Word Embeddings dengan AdaGrad bisa diberhentikan ketika nilai akumulasi *cost function* pada Persamaan 2.9 mencapai nilai seminimal mungkin.

Hasil akhir dari proses ini adalah dua Word Embeddings untuk setiap kata unik pada korpus, yaitu Word Embeddings untuk kata target dan Word Embeddings untuk kata konteks. Word Embeddings akhir yang digunakan adalah jumlah dari kedua Word Embeddings tersebut. Terakhir, seluruh kata dan Word Embeddings ini akan disimpan di dalam basis data.

3.2 Desain Aplikasi CNN

Pada tugas akhir ini, Convolutional Neural Networks (CNN) digunakan empat kali. Yaitu, pada Source Task, Target Task, satu CNN hasil transfer pada Transfer Learning Skenario 1, dan satu CNN hasil transfer pada Trasfer Learning Skenario 2 (lihat sub-bab 3.3). Semuanya menggunakan arsitektur CNN yang sama sehingga desain aplikasi CNN pada tugas akhir ini sama pada seluruh *task* yang melibatkan CNN.

Seperti yang diilustrasikan pada Gambar 3.3, langkah-langkah aplikasi CNN adalah sebagai berikut.



Gambar 3.4. *Mendapatkan Sentence Embeddings pada salah satu dokumen.*

3.2.1. *Text Preprocessing*

Setiap kalimat (dokumen) pada dataset yang ingin diklasifikasi melalui proses *text preprocessing*. Tahap *text preprocessing* ini terdiri dari: membuat huruf menjadi huruf kecil ata *lower case*, menghilangkan tanda baca, menghilangkan angka, menghilangkan karakter-karakter lain selain huruf, dan menjadikannya token-token dalam *array*.

Proses *text preprocessing* lain seperti mengubah setiap kata menjadi kata dasar (*stemming*), membuang kata yang sangat sering muncul (*stopwords*), atau menentukan unsur dari setiap kata (*part of speech tagging)* tidak dilakukan.

3.2.2. *Sentence Mapping (Average)*

Setelah setiap dokumen melalui tahap *text preprocessing*, kemudian dokumen-dokumen ini akan dicari bentuk representasi vektornya atau disebut dengan Sentence Embeddings. Proses ini disebut dengan *sentence mapping*. Representasi vektor dokumen ditemukan dengan cara merata-rata setiap Word Embeddings kata pembentuk dokumen. Setiap kata yang menyusun dokumen mempunyai Word Embeddings yang tersimpan di dalam basis data. Jika ada kata dalam dokumen yang tidak terdapat dalam basis data (tidak mempunyai Word Embeddings), maka kata tersebut tidak dianggap ada dalam dokumen. Proses ini secara rinci dijelaskan pada Persamaan 2.13. Proses ini diilustrasikan pada Gambar 3.4.

Ketika seluruh dokumen telah didapatkan representasi vektornya, kemudian dibentuk sebuah matriks yang menumpuk

(*stacking*) seluruh Sentence Embeddings dari setiap dokumen. Setiap label yang menjadi kelas dari masing-masing dokumen juga diubah bentuknya menjadi *one hot vector*. *One hot vector* adalah vektor yang berisi angka 0 sebanyak jumlah kelas yang ada dan berisi satu angka 1 yang menandakan kelas untuk sebuah dokumen. Kumpulan *one hot vector* ini juga ditumpuk menjadi sebuah matriks. Kedua matriks ini menjadi masukan dari CNN.

3.2.3. Pemisahan Dataset

Untuk dapat mengukur performa model, matriks kumpulan Sentence Embeddings beserta labelnya dipisah menjadi dua bagian, yaitu *data training* dan *data testing*. *Data training* adalah data yang menjadi bahan CNN untuk melakukan *learning*. Setelah proses *learning* selesai, model CNN terbaik akan di-*load* kembali untuk memprediksi dari setiap dokumen *data testing*.

Label dari setiap *test set* yang sudah berupa matriks *one hot vector* akan menjadi *groundtruth* dari hasil prediksi CNN.

3.2.4. CNN *Learning*

Sebelum melakukan proses *learning* dengan CNN, penting untuk menentukan arsitektur CNN itu sendiri. Arsitektur yang cocok dengan dataset akan memberikan performa terbaik. Untuk mendapatkan arsitektur terbaik, dilakukan berbagai macam eksperimen dari kombinasi parameter-parameter pada CNN itu sendiri. Hasil eksperimen dan penentuan setiap parameter, jumlah *neurons*, dan nilai variabel-variabel ini akan dijelaskan pada sub-bab 3.4. Layer-layer CNN yang digunakan pada tugas akhir ini adalah satu layer konvolusi satu dimensi, satu layer *hidden*, dan satu layer output sesuai dengan publikasi dari (Yoon Kim, 2014).

Dalam proses *training/learning* dengan menggunakan *neural networks*, seringkali model terbaik bukan didapatkan pada *epochs* paling terakhir. Maka dari itu, proses *learning* dengan CNN akan melibatkan *checkpoints* dan *early stopping*.

Checkpoints adalah teknik menyimpan pemodelan CNN setiap nilai *loss* turun pada selisih tertentu. Dengan begitu, ketika nilai

loss sudah konstan atau cenderung naik, model CNN yang berhasil menggapai nilai *loss* terendah akan tersimpan.

Early stopping adalah teknik untuk memberhentikan proses *learning* CNN ketika nilai *loss* sudah tidak menunjukan penurunan yang signifikan dalam jumlah *epochs* tertentu. Cara ini digunakan karena dapat mengoptimalkan jumlah *epochs* semaksimal mungkin, tetapi lebih menghemat waktu dengan memberhentikan *training* CNN ketika sudah tidak menunjukan peningkatan *learning*.

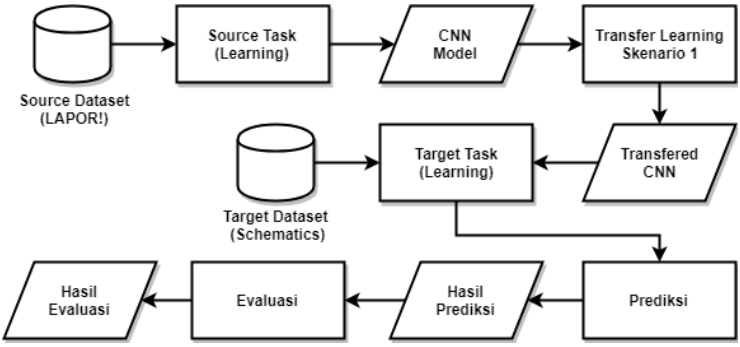
Salah satu algoritma penting dalam proses *learning* pada parameter-parameter *neural networks* adalah *Gradient Descent*. Algoritma inilah yang berperan dalam melakukan *update* pada bobot-bobot setiap *neurons*. Pada tugas akhir ini, algoritma *gradient descent* yang digunakan Adaptive Moment Estimation (Adam). Seperti yang dijabarkan pada sub-bab 2.6, Adam memberikan kecepatan *learning* yang tinggi pada CNN.

3.2.5. Prediksi

Langkah ini dilakukan untuk mengukur kemampuan model CNN terbaik dari hasil *training* sebelumnya dalam memprediksi. Model CNN ini diberi masukan dari *data testing.* Pada setiap Sentence Embeddings yang masuk, layer output dari CNN akan memberikan *probability* dari masing-masing kelas dengan rumus *softmax*. Sebuah neuron ke-i dengan nilai output *probability* tertinggi adalah hasil prediksi dari CNN.

3.2.6. Evaluasi

Setelah setiap Sentence Embeddings diprediksi kelasnya, langkah berikutnya adalah mengukur performa model. Pada tahap ini, hasil prediksi model dibandingkan dengan *groundtruth*. Agar mempermudah, dibuatlah sebuah *confusion matrix* untuk mendapatkan nilai akurasi, *recall*, *precision*, dan *F-Measure* dari setiap evaluasi model.



Gambar 3.5. *Desain GloVe, CNN, dan Transfer Learning Skenario 1.*

3.3 Desain Aplikasi Uji Coba GloVe dan CNN

Pada tugas akhir ini, akan ada tiga uji coba yang membandingkan performa model CNN. Penjabarannya adalah sebagai berikut.

3.3.1. GloVe dan CNN Tanpa Transfer Learning

Tujuan dari dibuatnya skenario kombinasi GloVe dan CNN tanpa Transfer Learning adalah untuk membandingkan apakah model GloVe dan CNN dengan Transfer Learning berhasil mengatasi permasalahan data latih awal yang relatif sedikit. Hal ini berdasarkan asumsi awal bahwa data FAQs Schematics yang didapatkan sangat sedikit untuk mencukupi kebutuhan CNN akan data yang besar.

Pada dasarnya, desain aplikasi GloVe dan CNN tanpa Transfer Learning sama persis dengan desain pada sub-bab 3.2. Karena tidak ada proses Transfer Learning pada aplikasi ini, desain tidak mengalami modifikasi. Aplikasi GloVe dan CNN tanpa Transfer Learning hanya menggunakan data FAQs Schematics sebagai masukannya.

3.3.2. GloVe, CNN, dan Transfer Learning Skenario 1

Penjelasan langkah-langkah pada desain GloVe, CNN, dan Transfer Learning Skenario 1 (Gambar 3.5) adalah sebagai berikut.

a. *Source Task* ($\mathcal{T}_S$)

Pada langkah ini, setiap neuron-neuron yang ada pada layer CNN diinisialisasi secara random. Dengan menggunakan dataset LAPOR!, CNN ini dilatih dengan langkah-langkah yang sama pada sub-bab 3.2. Prediksi dan evaluasi dari Source Task ini sebenarnya tidak akan menjadi salah satu hasil penelitian tugas akhir ini.

b. Transfer Learning Skenario 1

Pada proses Transfer Learning ini, model terbaik hasil dari $\mathcal{T}_S$ digunakan kembali. Layer yang diambil dari model $\mathcal{T}_S$ adalah layer konvolusinya saja. Karena menurut Yosinski et al. (2014) dan Semwal et al. (2018), layer pada *neural networks* yang lebih dekat pada input melakukan tugas yang lebih umum dibandingkan dengan layer yang lebih dekat pada output. Makin mendekati output, layer-layer tersebut semakin melakukan tugas yang spesifik sesuai dengan dataset. Maka dari itu, layer konvolusi pada $\mathcal{T}_S$ bisa digunakan karena melakukan tugas yang umum. Sedangkan, layer *hidden* dan layer output pada $\mathcal{T}_S$ tidak bisa ditransfer karena melakukan tugas yang spesifik pada dataset LAPOR!.

c. Target Task ($\mathcal{T}_T$)

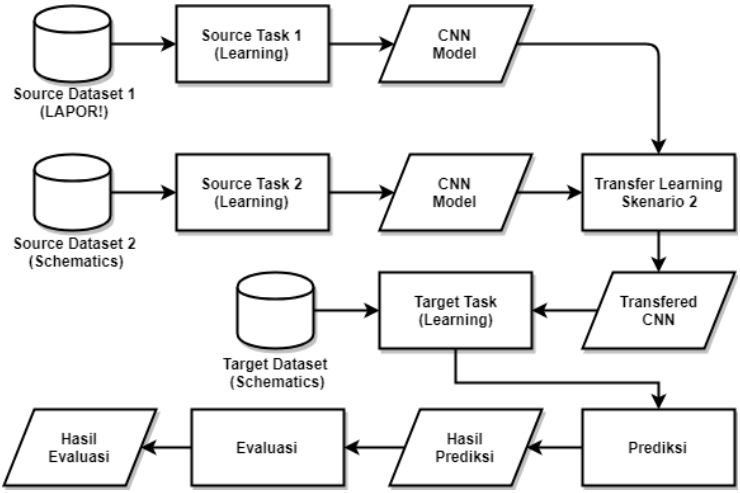
Model CNN pada $\mathcal{T}_T$ Transfer Learning Skenario 1 terdiri dari satu layer konvolusi hasil transfer dari $\mathcal{T}_S$, layer *hidden* yang diinisialisasi *random*, dan layer output yang diinisialisasi *random*. TT menggunakan dataset FAQs Schematics sebagai data latih.

d. Prediksi dan Evaluasi

Model terbaik hasil dari $\mathcal{T}_T$ digunakan untuk memprediksi *data testing* yang telah dipisahkan sebelumnya. Hasil prediksi dan *groundtruth*-nya digunakan untuk mendapatkan nilai akurasi, *precision*, *recall*, dan *F-Measure*.

3.3.3. GloVe, CNN, dan Transfer Learning Skenario 2

Penjelasan langkah-langkah pada desain GloVe, CNN, dan Transfer Learning Skenario 2 adalah sebagai berikut (Gambar 3.6).



Gambar 3.6. *Desain GloVe, CNN, dan Transfer Learning Skenario 2*

a. Source Task 1 ($\mathcal{T}_S$-1)

Pada langkah ini, setiap neuron-neuron yang ada pada layer CNN diinisialisasi secara random. Dengan menggunakan dataset LAPOR!, CNN ini dilatih dengan langkah-langkah yang sama pada sub-bab 3.2. Prediksi dan evaluasi dari Source Task ini sebenarnya tidak akan menjadi salah satu hasil penelitian tugas akhir ini.

b. Source Task 2 ($\mathcal{T}_S$-2)

Pada langkah ini, setiap neuron-neuron yang ada pada layer CNN diinisialisasi secara random. Dengan menggunakan dataset FAQs Schematics, CNN ini dilatih dengan langkah-langkah yang sama pada sub-bab 3.2. Prediksi dan evaluasi dari Source Task ini sebenarnya tidak akan menjadi salah satu hasil penelitian tugas akhir ini.

c. Transfer Learning Skenario 2

Pada proses Transfer Learning ini, layer-layer model pada $\mathcal{T}_S$-1 dan $\mathcal{T}_S$-2 digunakan kembali. Gabungan kedua model ini akan

membentuk CNN yang kemudian akan dilatih kembali dengan data FAQs Schematics.

Jadi, CNN pada $\mathcal{T}_T$ akan terdiri dari dua gabungan layer hasil transfer $\mathcal{T}_S$-1 dan $\mathcal{T}_S$-2. Layer konvolusi pada $\mathcal{T}_T$ akan menggunakan layer konvolusi dari $\mathcal{T}_S$-1. Layer *hidden* dan layer output pada TT akan menggunakan layer *hidden* dan layer output hasil dari $\mathcal{T}_S$-2. Ini dikarenakan layer konvolusi pada $\mathcal{T}_S$-1 sudah pernah melakukan tugas umum dengan data yang jauh lebih besar. Layer *hidden* dan layer output pada $\mathcal{T}_S$-2 digunakan kembali karena pernah melakukan tugas spesifik pada data FAQs Schematics. Gabungan transfer kedua model ini diharapkan memberikan performa yang lebih baik.

d. Target Task ($\mathcal{T}_T$)

Seperti yang sudah dijelaskan sebelumnya, model CNN pada TT akan menggunakan gabungan dari dua model Source Task sebelumnya.

e. Prediksi dan Evaluasi

Model terbaik hasil dari $\mathcal{T}_T$ digunakan untuk memprediksi *data testing* yang telah dipisahkan sebelumnya. Hasil prediksi dan *groundtruth*-nya digunakan untuk mendapatkan nilai akurasi, *precision*, *recall*, dan *F-Measure*.

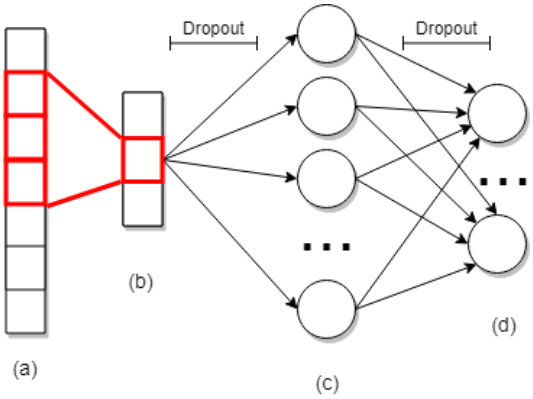
3.4 Arsitektur CNN

Pada sub-bab ini akan dijelaskan desain arsitektur CNN.

3.4.1. Eksperimen Arsitektur CNN

Arsitektur CNN yang digunakan sangat berpengaruh sekali dalam performa klasifikasi. Maka dari itu, untuk menentukan jumlah layer, parameter, atau nilai variabel dalam CNN perlu melalui eksperimen.

Untuk jumlah layer yang digunakan, tugas akhir ini merujuk pada publikasi Yoon Kim (2014) yang berjudul "*Convolutional Neural Networks for Sentence Classification"*. Arsitektur yang



Gambar 3.7. *Arsitektur CNN secara umum yang terdiri dari (a) Sentence Embeddings, (b) convolutional layer, (b) hidden layer, dan (c) output layer.*

digunakan adalah satu *convolutional layer*, satu *hidden layer*, dan satu *output layer*. Layer konvolusi akan menerima masukan berupa Sentence Embeddings hasil dari merata-rata Word Embeddings pada kata yang menyusun dokumen. Proses ini digambarkan pada Gambar 3.4. Regularisasi Dropout (Srivastava et al., 2014) diterapkan di antara layer konvolusi-layer *hidden* dan di antara layer *hidden*-layer output. Jadi, arsitektur CNN secara umum dalam tugas akhir ini seperti pada Gambar 3.7. Fungsi aktivasi ReLu, *sigmoid*, dan *softmax* masing-masing digunakan pada layer konvolusi, layer *hidden*, dan layer output.

Untuk menentukan jumlah neuron pada setiap layer, dilakukan eksperimen CNN dengan menggunakan dataset FAQs Schematics. Rentang nilai kernel layer konvolusi yang diuji coba adalah 3-10. Jumlah neuron pada layer konvolusi dan layer *hidden* yang diuji coba ada pada rentang 10-500 dan 16-180. Sedangkan untuk layer output, jumlah neuronnya mengikuti jumlah kelas pada dataset. Yaitu, 68 untuk dataset LAPOR! dan 22 untuk dataset FAQs Schematics. Hasil eksperimen pada berbagai macam jumlah neuron dan *dropout* dapat dilihat pada Tabel 3.1.

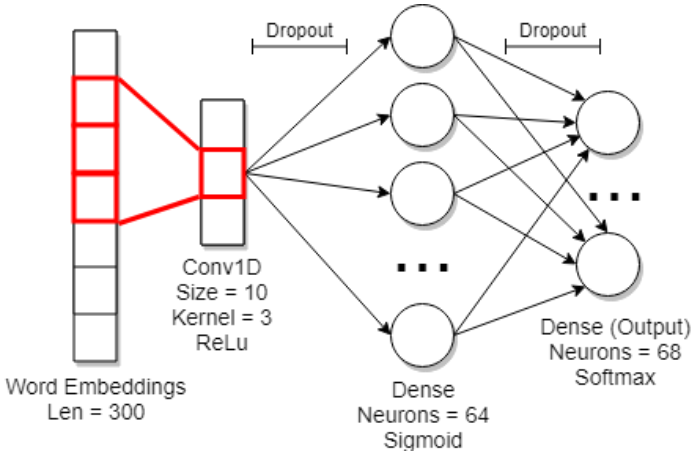
Tabel 3.1. *Eksperimen arsitektur CNN pada dataset FAQs Schematics.*

No.	Kernel	Conv1D	Hidden	Dropout1	Dropout2	Acc (%)	Pre (%)	Re (%)	F1 (%)	Loss
1.	3	10	16	0,1	0,1	90,5	91,5	90,5	90,2	0,038
2.	3	10	32	0,1	0,1	92,2	93,1	92,2	92,2	0,025
3.	**3**	**10**	**64**	**0,1**	**0,1**	**94,4**	**95,2**	**94,4**	**94,3**	**0,018**
4.	3	10	128	0,1	0,1	93,9	94,8	93,3	93,9	0,018
5.	5	10	64	0,1	0,1	93,1	93,7	93,1	93,1	0,019
6.	3	50	64	0,1	0,1	92,2	92,9	92,2	92,2	0,021
7.	10	50	64	0,1	0,1	93,5	94,2	93,5	93,4	0,018
8.	10	100	64	0,1	0,1	93,9	94,7	93,9	93,9	0,016
9.	10	500	64	0,1	0,1	91,2	92,4	91,2	91,7	0,023
10.	10	500	128	0,1	0,1	92,2	92,9	92,2	92,2	0,018
11.	10	500	180	0,2	0,2	92,2	92,3	92,2	92,2	0,019
12.	3	10	64	0,2	0,2	91,2	92,5	91,8	91,7	0,021
13.	3	10	64	0,1	0,5	91,2	92,9	91,8	91,7	0,027
14.	3	5	64	0,1	0,1	93,9	94,5	93,9	93,9	0,021
15.	2	5	64	0,1	0,1	89,6	90,5	89,6	89,5	0,022
16.	1	5	64	0,1	0,1	93,1	93,8	93,1	93	0,02

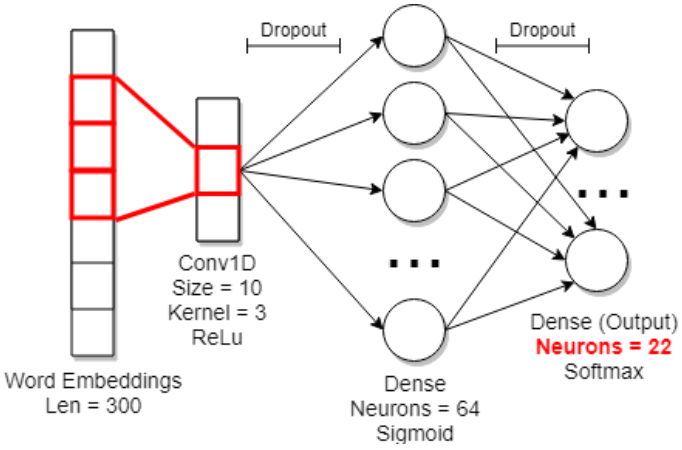
Arsitektur terbesar yang dicoba ada pada percobaan ke-11 dengan besar kernal 10, neuron pada layer konvolusi 500, neuron pada layer *hidden* 180, dan nilai kedua *dropout* 0.2. Karena keterbatasan *resource*, arsitektur tersebut adalah arsitektur terbesar yang bisa dicoba (dengan GPU 3GB Nvidia GeForce GTX 1060). Meski begitu, arsitektur CNN yang memberikan performa terbaik ada pada percobaan ke-3 dengan besar kernel 3, neuron layer konvolusi 10, neuron pada layer *hidden* 64, dan nilai masing-masing *dropout* 0,1. Arsitektur ini memberikan nilai akurasi, *precision*, *recall*, *F-Measure*, dan *loss* masing-masing sebesar 94,4%; 95,2%; 94,4%; 94,3%; dan 0,018. Maka dari itu, arsitektur inilah yang akan digunakan pada seluruh skenario dalam tugas akhir ini.

3.4.2. Arsitektur CNN Pada Dataset LAPOR!

Sesuai dengan hasil dari eksperimen pada sub-bab 3.4.1, arsitektur CNN yang digunakan memiliki kernel 3 pada layer konvolusi, neuron 10 dan 64 pada layer konvolusi dan layer *hidden*, dan nilai kedua *dropout* adalah 0,1. Sedangkan, layer output pada CNN akan memiliki 68 neuron. Ini disebabkan jumlah kelas pada



Gambar 3.8. *Arsitektur CNN yang digunakan pada dataset LAPOR!*



***Gambar 3.9**. Arsitektur CNN yang digunakan pada dataset FAQs Schematics dengan output layer 22.*

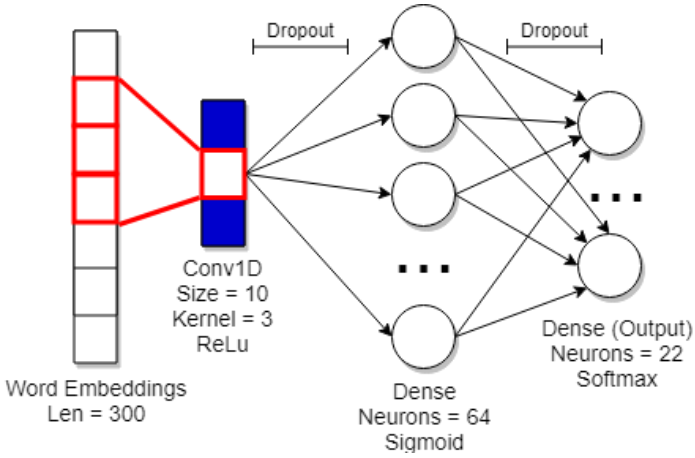
dataset LAPOR! ada 68 kelas. Ilustrasi arsitektur CNN ini ada pada Gambar 3.8.

3.4.3. Arsitektur CNN Pada Dataset FAQs Schematics

Arsitektur CNN yang digunakan pada dataset FAQs Schematics sama persis dengan arsitektur yang digunakan CNN pada sub-bab 3.4.2. Perbedaannya hanya pada jumlah neuron pada layer output yang berjumlah 22. Ini disebabkan kelas pada data FAQs Schematics berjumlah 22. Ilustrasi arsitektur CNN pada Dataset FAQs Schematics dapat dilihat pada Gambar 3.9.

3.4.4. Arsitektur CNN pada Transfer Learning Skenario 1

Arsitektur CNN pada Transfer Learning Skenario 2 sebenarnya mempunyai arsitektur yang sama dengan arsitektur yang digunakan pada sub-bab 3.4.3 (karena dilatih dengan menggunakan data FAQs Schematics). Hanya saja mempunyai perbedaan pada layer konvolusi. Layer konvolusi yang digunakan adalah layer konvolusi hasil *training* CNN yang dilakukan pada



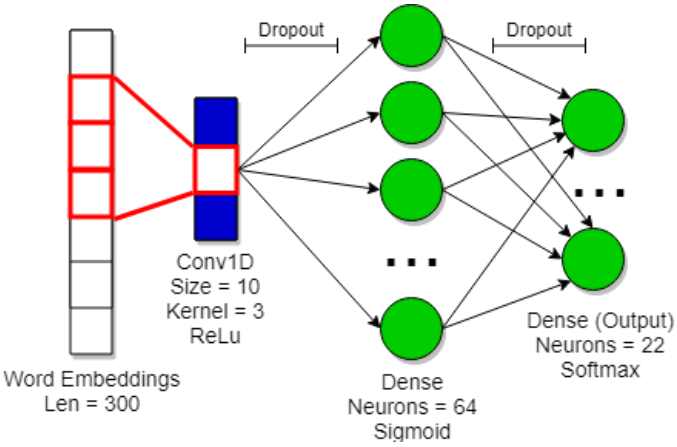
***Gambar 3.10**. Arsitektur CNN yang digunakan pada Transfer Learning Skenario 1. Layer konvolusi (berwarna biru) merupakan hasil transfer dari CNN yang dilatih dengan data LAPOR!*

sub-bab 3.4.2. Jadi, layer konvolusi ditransfer ke CNN yang digunakan pada arsitektur ini. Ilustrasi CNN yang digunakan pada Transfer Learning skenario 1 ada pada Gambar 3.10.

Mengapa hanya layer konvolusi yang ditransfer? Penjelasan ini ada pada sub-bab 3.3.2. Melakukan transfer pada layer *hidden* dan layer output pada CNN yang melakukan klasifikasi pada dataset yang berbeda ternyata memberikan performa yang lebih buruk (Semwal et al., 2018).

3.4.5. Arsitektur CNN pada Transfer Learning Skenario 2

Arsitektur CNN yang digunakan pada Transfer Learning Skenario 2 adalah gabungan dari arsitektur CNN yang digunakan pada sub-bab 3.4.2 dan 3.4.3. Layer konvolusi pada CNN ini menggunakan layer konvolusi yang sama dengan CNN yang digunakan pada sub-bab 3.4.2. Sedangkan layer *hidden* dan layer output yang digunakan pada CNN ini menggunakan layer yang sama dengan CNN yang digunakan pada sub-bab 3.4.3. Alasan

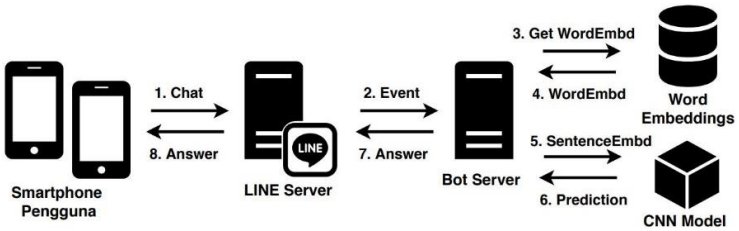


Gambar 3.11. *Arsitektur CNN yang digunakan pada Transfer Learning Skenario 2. Layer konvolusi (berwarna biru) merupakan hasil transfer dari CNN yang dilatih dengan data LAPOR! sedangkan layer hidden dan layer output (berwarna hijau) merupakan hasil transfer dari CNN yang dilatih dengan data FAQs Schematics.*

dibalik pemilihan transfer layer seperti ini ada pada sub-bab 3.3.3. Ilustrasi CNN yang digunakan pada Transfer Learning Skenario 2 ada pada Gambar 3.11.

3.5 Desain Aplikasi Chatbot

Setelah mendapatkan model dengan performa terbaik hasil dari Transfer Learning, kemudian model ini akan digunakan untuk memprediksi kelas dari kalimat-kalimat yang masuk pada sistem *chatbot*. Pada tugas akhir ini, digunakan LINE Messenger API (*Application Program Interface*) sehingga interaksi dengan *chatbot* akan berlangsung melalui aplikasi LINE. Untuk memberikan personifikasi yang baik pada *chatbot*, *bot* yang digunakan diberi nama Mira. Mira akan memeberikan jawaban dari *frequently asked question* seputar kegiatan Schematics HMTC ITS. Gambar 3.12 menjelaskan bagaimana arsitektur aplikasi *chatbot* diimplementasikan hingga jawaban dari perintah atau pertanyaan dari *chat* pengguna diprediksi dan dijawab oleh *bot*. *Chat* yang



Gambar 3.12. *Alur komunikasi antar end-point pada aplikasi chatbots.*

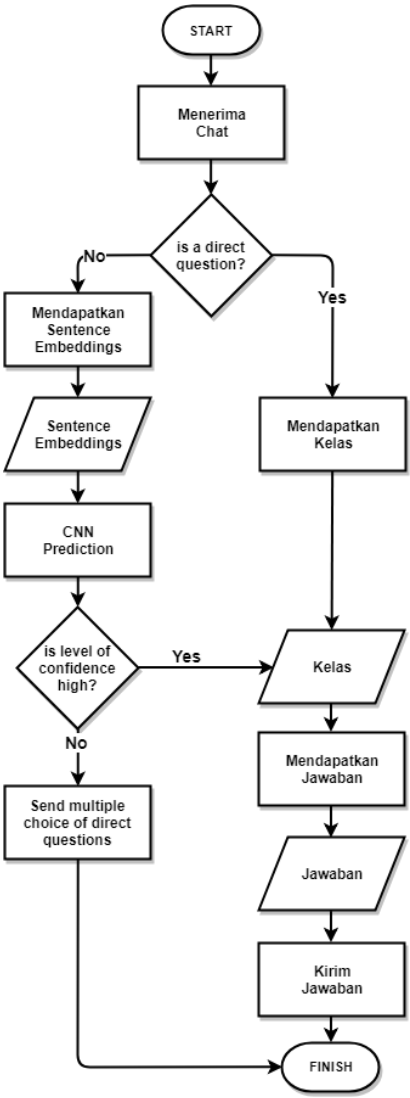
dikirim melalui aplikasi LINE dari *smartphone* akan diikirimkan ke server LINE. Dengan menggunakan *webhook*, server LINE akan mengirimkannya ke server *bot* berupa sebuah *event*. Isi dari *event* adalah berbagai macam data pengirim, termasuk nama pengirim, isi *chat*, waktu *chat*, dan lainnya.

Gambar 3.13 menunjukan bagaimana sistem *chatbot* pada tugas akhir ini mengani *chat* masuk. *Chat* yang masuk akan diperiksa apakah termasuk *direct question*. Jika ya, maka kelas dari pertanyaan bisa langsung diprediksi dan dikirimkan jawabannya. Jika bukan, *chat* tersebut akan menjadi dokumen yang dikonversi menjadi Sentence Embeddings. Kemudian, Sentence Embeddings ini akan menjadi masukan model CNN terbaik hasil dari uji coba pada tugas akhir ini. Setelah hasil dari *softmax function* didapatkan, akan dicek terlebih dahulu kelas dengan *probability* tertinggi. Jika nilainya tidak melewati ambang batas tertentu (*threshold*), maka sistem akan mengirimkan *multiple choice* dari *direct questions* kepada pengguna sehingga pengguna bisa memilih pertanyaan yang bisa dijawab oleh sistem. Jika nilai *probabilty* tertingginya melewati *threshold*, maka kelas tersebut akan dijadikan hasil prediksi untuk mendapatkan jawabannya. Jawaban yang didapat kemudian dikirimkan ke pengguna.

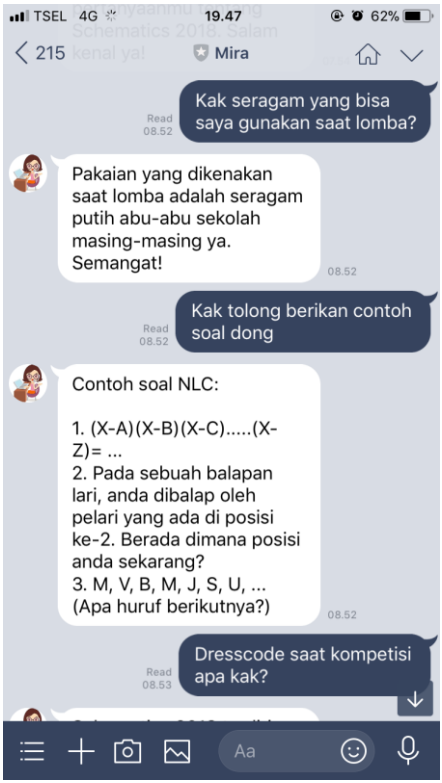
3.6 Desain *User Interface* Pada *Chatbot*

Desain *user interface* pada *chatbot* yang didesain pada tugas akhir ini sejatinya adalah *conversational interface*. Karena pada tugas akhir ini digunakan LINE API Messenger sebagai platform *chat*, antarmuka pengguna yang digunakan adalah fitur-fitur

conversational interface yang sudah disediakan oleh LINE API Messenger. Rinciannya adalah sebagaimana berikut.



Gambar 3.13. *Flowchart sistem bot dalam menangani chat.*



Gambar 3.14. *Desain user interface chat pada LINE Messenger*.

3.6.1. Chat

Chat adalah fitur standar yang ada pada LINE Messenger. Seperti yang dapat dilihat pada Gambar 3.13, setiap skenario *chat* yang masuk pada sistem akan dibalas oleh sistem dengan sebuah jawaban. Desain antarmuka *chat* yang digunakan dapat dilihat pada Gambar 3.14.

3.6.2. Pertanyaan Langsung (*Direct Question*)

Direct Question adalah *chat* yang tidak memerlukan prediksi CNN untuk memberikan jawaban ke pengguna. Jika sebuah *chat*

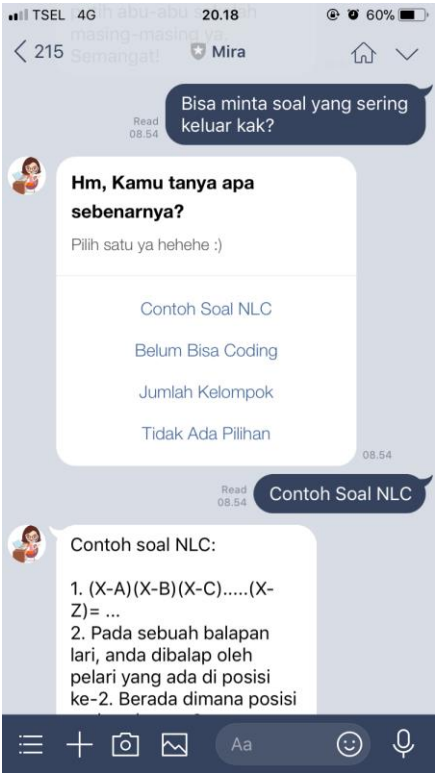
masuk dan diidentifikasi sebagai *direct question* oleh sistem, maka sistem langsung mengirimkan jawaban kepada pengguna (lihat *flowchart* pada Gambar 3.13). Teknik ini digunakan untuk mempermudah sistem ketika pengguna mengirimkan pertanyaan ambigu. Penjelasan rinci tentang pertanyaan ambigu dan bagaimana mengatasinya akan dijelaskan pada sub-bab 3.6.3.

3.6.3. *Multiple Choices of Direct Question*

Ketika CNN tidak memberikan probabilitas yang tinggi pada kelas dengan probabilitas tertinggi, maka sistem akan memutuskan

Tabel 3. 2. *Direct Question beserta kelasnya.*

Direct Question	Kelas
Greetings	Greetings
Apa Itu Schematics	What is Schematics
Apa Itu NLC	What is NLC
Apa Itu NPC	What is NPC
Keuntungan NLC	Benefit NLC
Keuntungan NPC	Benefit NPC
Contoh Soal NLC	Soal NLC
Belum Bisa Coding	Cannot Code Yet
Di mana & Kapan NLC	Where When NLC
Di mana & Kapan NPC	Where When NPC
Daftar NLC & NPC	Pendaftaran NLC NPC
Timeline NLC	Timeline NLC
Timeline NPC	Timeline NPC
Biaya NLC	Biaya NLC
Biaya NPC	Biaya NPC
Verifikasi Daftar	Belum Verifikasi
Pindah Region NLC	Pindah Region NLC
Jumlah Kelompok	Jumlah Kelompok
Tentang Sertifikat	Sertifikat Tidak Datang
Tiket NST & Reeva	Tiket NST Reeva
Pengumuman NLC&NPC	Pengumuman NLC NPC
Dresscode Lomba	Aturan Seragam
Tidak Ada Pilihan	-



Gambar 3.15. *Desain antarmuka multiple choices of direct questions. Terlihat pada chat selanjutnya, pengguna memilih direct question "Contoh Soal NLC".*

bahwa pertanyaan ini adalah pertanyaan yang ambigu (lihat Gambar 3.13). Sistem akan memilih 3 kelas tertinggi dan memberikan *multiple choices* pada pengguna berupa *direct questions* yang setiap pilihannya bisa dijawab langsung oleh sistem *chatbot*. Pilihan keempat adalah "Tidak Ada Pilihan" yang menandakan tiga pilihan kelas dengan prediksi tertinggi juga bukan jawaban yang diinginkan pengguna. Penampakan antarmuka *multiple choice of direct question* ada pada Gambar 3.15.

3.6.4. Jawaban

Baik pada skenario *direct questions* ataupun skenario dengan prediksi CNN, sistem *chatbot* akan memberikan jawaban pada setiap *chat* pengguna yang masuk. Pada tugas akhir ini, aplikasi *chatbot* menyediakan satu jawaban pada setiap kelas. Rincian jawaban pada masing-masing kelas bisa dilihat pada Lampiran 1.

[*Halaman ini sengaja dikosongkan*]

BAB IV
IMPLEMENTASI

Pada bab ini akan dijelaskan implementasi pada tugas akhir ini. Bab ini juga akan merinci *tools* yang digunakan pada tugas akhir ini beserta langkah-langkah pengerjaannya.

4.1 Tools

Pada Tabel 4.1 diuraikan *tools* yang digunakan untuk pengerjaan tugas akhir ini.

Tabel 4.1. *tools yang digunakan pada tugas akhir.*

No.	Tools	Deskripsi
1	C	Bahasa C digunakan menangani korpus GloVe yang besar.
2	Python	Bahasa Python digunakan untuk menangani *task* Natural Language Processing (NLP) dan Machine Learning.
3	Gensim	*Library* yang digunakan untuk melakukan *text preprocessing*.
4	MySQL	Menyimpan hasil Word Embeddings di MySQL DB.
5	TensorFlow	*Library* yang digunakan untuk melakukan training menggunakan Neural Networks.
6	Keras	*Library* yang digunakan untuk menyusun tensor-tensor dari TensorFlow sehingga lebih mudah dalam membangun arsitektur CNN.
7	Scikit-Learn	Library ini digunakan untuk melakukan evaluasi model. Baik *accuracy*, *precision*, *recall*, maupun *F-Measure*.

4.2 Tahap-Tahap Implementasi

Tahap-tahap implementasi pada tugas akhir ini secara garis besar ada 4 tahap: mendapatkan Word Embeddings dengan GloVe; implementasi CNN; implementasi GloVe dan CNN; implementasi GloVe, CNN, dan Transfer Learning Skenario 1; dan implementasi GloVe, CNN, dan Transfer Learning Skenario 2.

4.3 Mendapatkan Word Embeddings dengan GloVe

Pada sub-bab ini akan dijelaskan secara rinci tahap-tahap implementasi mendapatkan Word Embeddings dengan algoritma GloVe.

4.3.1. Korpus

Korpus yang digunakan pada tugas akhir ini adalah Wikipedia Bahasa Indonesia yang dipublikasi secara terbuka di alamat https://dumps.wikimedia.org/idwiki/latest dan dokumen-dokumen pada data LAPOR! yang dipublikasi pemerintah Indonesia di *https://lapor.go.id*. Total token yang digunakan adalah sejumlah 83 juta token dengan 2.216.260 kata unik. Kata yang digunakan pada proses *training* adalah kata yang muncul paling sedikit 5 kali dalam korpus. Dengan begitu, ada 331.286 kata unik dalam korpus yang digunakan.

Setelah seluruh korpus Wikipedia Bahasa Indonesia diunduh, korpus ini kemudian melewati proses *parsing* dari format asalnya, yaitu XML. Hanya bagian isi artikel saja yang diambil untuk menjadi corpus. Proses *parsing* ini dilakukan dengan menggunakan *library* Gensim. Proses ini memakan waktu hampir 8 jam dengan menggunakan fungsi `gensim.corpora.WikiCorpus()`. Proses ini dapat dilihat pada Pseudocode 4.1.

Kelas `WikiCorpus` akan melakukan *parsing* pada selutuh artikel Wikipedia Bahasa Indonesia. Fungsi `get_text` pada Gensim

```
wiki_xml_file = letak/nama file korpus Wikipedia
corpus_output = letak/nama file output korpus

wiki = WikiCorpus(input=wiki_xml_file,
                  lemmatize=False, lower=True
                  tokenizer_func=my_tokenizer)

for article in wiki.get_texts :
   text = my_tokenizer(article)
   corpus_output.write(text + '\n')
```

Pseudocode 4.1. *Proses parsing korpus Wikipedia Bahasa Indonesia.*

```
1   docs = array dari dokumen dataset LAPOR!
2   corpus_output = letak/nama file output korpus
3
4   for doc in docs:
5      text = my_tokenizer(doc)
6      corpus_output.write(text + '\n')
```

Pseudocode 4.2. *Proses dokumen-dokumen LAPOR! menjadi korpus.*

berfungsi untuk mengambil konten artikel dari setiap artikel di Wikipedia Bahasa Indonesia. Sebelum menulis hasilnya pada sebuah file, setiap artikel akan melalui proses *text preprocessing* terlebih dahulu yang ada pada fungsi `my_tokenizer` (lihat sub-bab 4.2.6). Hasilnya kemudian ditulis ke dalam sebuah file dengan dipisahkan oleh *break line*.

Penanganan pada korpus LAPOR! bisa dilihat pada Pseudocode 4.2. Berbeda dengan Wikipedia Berbahasa Indonesia yang berformat XML, dokumen-dokumen pada data LAPOR! sudah dalam format CSV sehingga bisa dengan mudah membentuk array pada dokumen-dokumennya. Pada bari ke-4 sampai ke-6, setiap dokumen pada dataset LAPOR! diiterasi. Setiap dokumen tersebut melewati proses *text preprocessing* melalui fungsi `my_tokenizer`. Kemudian hasil ditulis pada file yang sama dengan hasil korpus Wikipedia Bahasa Indonesia.

4.3.2. *Text Preprocessing* Pada Korpus

Proses *text preprocessing* yang terjadi pada korpus dapat ditinjau dari Pseudocode 4.3. Fungsi ini menerima satu parameter yang dinamai `text`. Pada baris ke-2, setiap kata yang berpisah dengan spasi dipisahkan menjadi elemen-elemen pada array. Baris ke-7 sampai ke-10 adalah iterasi pada setiap kata yang menjadi elemen pada array `txt_space_splitted`. Setiap kata dihilangkan unsur non-hurufnya. Kecuali, karakter *dash*. Karena dalam bahasa Indonesia ada kata yang disebut berulang (disambung dengan karakter *dash*) dan dianggap menjadi satu kata. Seperti, kupu-kupu, ubur-ubur, dan lainnya. Setelah itu, setiap kata diubah menjadi

```
1   function my_tokenizer(text):
2     txt_space_splitted = text.split(space)
3     remover = regular_expression("[^a-zA-Z-]")
4
5     tokens = inisialisasi array dari token-token
6
7     for word in txt_space_splitted:
8       term = remover.sub('', word)
9       term = term.lower
10      tokens.append(term)
11
12    return tokens
```

Pseudocode 4.3. *Kode text preprocessing pada korpus.*

lower case. Terakhir, kata-kata ini ditambahkan ke dalam *array* yang menyimpan kata-kata yang telah melalui *text preprocessing*.

4.3.3. Membangun *Word-Word Co-Occurrence Matrix*

Sebelum melatih parameter Word Embeddings dengan menggunakan AdaGrad harus dibangun terlebih dahulu matriks *word-word co-occurrence*. Secara rinci, kode dalam membangun matriks *word-word co-occurrence* ada pada Pseudocode 4.4.

Matriks dibangun dengan cara memindai dokumen pada setiap korpus yang dipisahkan dengan karakter *break line*. Lalu, setiap kata yang ada pada dokumen dicari nomor indeksnya (nomor indeks ini bisa didapatkan dari mengurutkan seluruh kata unik yang ada pada korpus). Proses perulangan ini ada pada baris ke-6 sampai ke-20. Pada baris ke-10 sampai 13, kode ini berfungsi untuk mendapatkan nomor indeks pada kata konteks (kata yang masuk pada rentang *window* sebesar 10). Baris ke-15 sampai 20, pada setiap kata konteks dihitung nilai yang akan mengisi elemen pada matriks (baris ke-16 dan ke-17) lalu menambahkannya pada elemen yang benar (baris ke-19 dan ke-20).

4.3.4. AdaGrad *Training*

Untuk membahas kode yang menjalankan algoritma AdaGrad perlu untuk mengetahui terlebih dahulu nama-nama variabel yang

```
1   cooccur = Inisialisasi word-word co-occurrence matrix
2   vocab = berisi kata unik yang ada pada korpus
3   vocab_size = jumlah token unik pada korpus
4   window = panjang window untuk memindai korpus
5
6   for line in corpus:
7     tokens = line.split(space)
8     token_ids = [vocab[word].index for word in tokens]
9
10    for target_i, target_id in enumerate token_ids:
11      context_ids = token_ids[max(0, center_i - window)
12                                : center_i]
13      contextd_len = len(context_ids)
14
15      for left_i, left_id in enumerate(context_ids):
16        distance = contexts_len – left_i
17        increment = 1 / distance
18
19        cooccur[center_id, left_id] += increment
20        cooccur[left_id, center_id] += increment
```

Pseudocode 4.4. *Kode membangun matriks word-word co-occurrence.*

digunakan pada Pseudocode 4.4. Dilihat pada Pseudocode 4.4, `grad_w_main`, `grad_w_context`, `grad_bias_main`, `grad_bias_context` masing-masing adalah variabel yang menampung nilai gradien dari vektor kata target, vektor kata konteks, bias kata target, dan bias kata konteks. Pada awalnya, seluruh nilai variabel tersebut diinisialisasi secara random. Variabel `vector_w_main`, `vector_w_context`, `bias_w_main`, dan `bias_w_context` masing-masing adalah variabel yang menambung nilai vektor kata target, vektor kata konteks, bias kata target, dan bias kata konteks. Variabel `grad_sq_w_main`, `grad_sq_w_context`, `grad_sq_b_main`, dan `grad_sq_b_context` masing-masing adalah variabel yang menampung jumlah kuadrat dari historis gradien vektor kata target, vektor kata konteks, bias kata target, dan bias kata konteks.

Pseudocode 4.4 juga menjelaskan bagian penting dari GloVe, yaitu proses *training* dengan AdaGrad. Dimulai dari mencari nilai gradien dari setiap parameter w_i, $\widetilde{w}_j$, b_i, dan b_j (baris 7-10). Lalu,

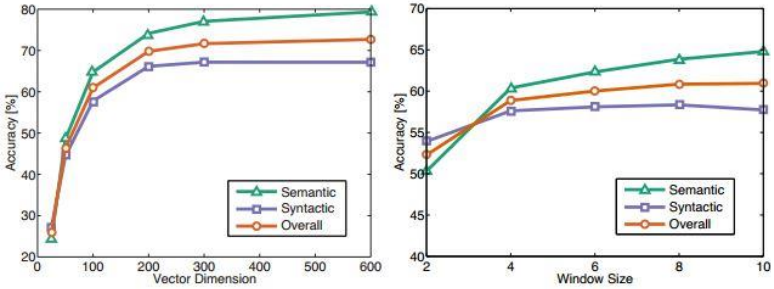
```
1   weight = f(X_ij)
2   cost_inner = (w_i^T w̃_j + b_i + b̃_j − logX_ij)
3
4   dj = 2 * weight * cost_inner
5
6   # mencari gradien dari setiap parameter
7   grad_w_main = vector_w_context * dj
8   grad_w_context = vector_w_main * dj
9   grad_bias_main = dj
10  grad_bias_context = dj
11
12  # update setiap parameter
13  vector_w_main -= (learning_rate * grad_w_main
14                    sqrt(grad_sq_w_main))
15  vector_w_context -= (learning_rate * grad_w_context
16                    sqrt(grad_sq_w_context))
17  bias_w_main -= (learning_rate * grad_b_main
18                    sqrt(grad_sq_b_main))
19  bias_w_context -= (learning_rate * grad_b_context
20                    sqrt(grad_sq_b_context))
21
22  # update jumlah kuadrat gradien
23  grad_sq_w_main += square(grad_w_main)
24  grad_sq_w_context += square(grad_w_context)
25  grad_sq_b_main += grad_b_main ** 2
26  grad_sq_b_context += grad_b_context ** 2
27
```

Pseudocode 4.5. *AdaGrad training pada GloVe.*

dilakukan update pada setiap parameter ini berdasarkan persamaan yang diusulkan oleh algoritma AdaGrad (baris 13-20). Terakhir, variabel yang menampung penjumlahan kuadrat gradien pada setiap parameter ditambahkan oleh kuadrat dari gradien setiap parameter yang baru saja didapatkan (baris 23-26).

Pennington et al. (2014) menetapkan nilai $x_{max} = 100$ dan $\alpha = 3/4$ pada setiap eksperimennya. Pada tugas akhir ini, panjang vektor yang ditentukan adalah 300 dimensi vektor karena jumlah ini memberikan hasil yang terbaik (Gambar 4.1).

Model ini kemudian dilatih dengan algoritma AdaGrad, *stochastically sampling* pada elemen bukan nol pada matriks X.



Gambar 4.1. *Vektor kata sebesar 300 dimensi dan window sebesar 10 memberikan hasil yang paling baik (Pennington et al., 2014).*

Jumlah iterasi yang ditentukan adalah 100 iterasi pada saat dilakukan proses *training* dengan AdaGrad. Algoritma AdaGrad dijelaskan secara rinci pada sub-bab 2.4.

4.4 Implementasi Pengujian GloVe Word Embeddings

Pada sub-bab ini akan dijelaskan implementasi program untuk menguji hasil dari Word Embeddings yang dihasilkan dari proses yang selesai pada sub-bab 4.3.

4.4.1. Uji Fungsionalitas Semantik GloVe Word Embeddings

Uji fungsionalitas semantik GloVe Word Embeddings dilakukan dengan cara memberikan data uji sintetis. Satu data uji sintetis terdiri dari 4 kata. Kata pertama and ketiga adalah nama ibukota negara dari kata kedua dan keempat. Secara rinci, penjelasan pada uji coba ini akan dijelaskan pada sub-bab 5.1.2.

Implementasi kode pengujian ini terdapat pada Pseudocode 4.6. Baris ke-4 adalah perulangan pada setiap data uji sintetis yang tersimpan dalam *array* `syntetic_data`. Baris ke-5 berfungsi untuk mendapatkan sebuah vektor hasil dari operasi pengurangan vektor dari kata pertama dengan kata kedua dan hasilnya ditambahkan dengan vektor dari kata ketiga. Hasil operasi vektor ini disimpan dalam variabel `vec`. Kemudian, vektor yang tersimpan di variabel `vec` dicari nilai *cosine similarity*-nya pada setiap vektor kata yang

```
1   syntetic_data = array data sintetis uji semantik
2   wordembd = vektor kata dari seluruh isi korpus
3
4   for word1, word2, word3, word4 in syntetic_data:
5     vec = word1.vec – word2.vec + word3.vec
6     dist = cosine_similarity(wordembd, vec)
7     predictions.append(dist.argmax)
8     groundtruth.append(word4.index)
9
10  val = predictions == groundtruth
11  count_total = len(question_data)
12  count_correct = [1 for x in val if x is True]
13
14  acc = count_correct / count_total
```

Pseudocode 4.6. *Kode uji fungsionalitas semantik GloVe Word Embeddings.*

ada pada korpus. Seluruh nilai *cosine similarity* ini disimpan pada variabel `dist` (baris ke-6). Pada baris ke-7, sebuah *array* `predictions` berfungsi menampung indeks kata yang memiliki nilai *cosine similarity* tertinggi dengan vektor `vec`. *Array* `groundtruth` menyimpan indeks kata yang merupakan kata keempat pada data uji sintetis. Pada baris ke-10, seluruh angka-angka indeks ini dibandingkan untuk diketahui berapa jumlah prediksi yang benar (baris ke 10-12). Terakhir, pada baris ke-14, didapatkan nilai akurasi prediksi.

4.4.2. Uji 10 Kata Terdekat GloVe Word Embeddings

Beragamnya jenis calon peserta yang bertanya membuat penggunaan kata tertentu juga beragam. Misal, beberapa calon peserta lebih memilih menggunakan kata “seragam” dibandingkan “baju” atau “pakaian”. Padahal, kata ini sebenarnya bermakna mirip. Sistem *chatbot* harus mampu menangani kasus ini. Untuk mengetahui apakah *chatbot* dapat mengintepretasikan beragam kalimat berkonteks sama ini mempunyai makna yang mirip, uji coba ini dilakukan untuk mencari 10 kata terdekat dalam ruang spasial Word Embeddings pada beberapa kata kunci yang sering

```
1   wordembd = vektor kata dari seluruh isi korpus
2
3   function get_10_nearest(word):
4     vector = word.vec
5     dist = cosine_similarity(wordembd, vec)
6     dist[dist.argmax] = -1
7       for i in 10:
8       highest = dist.argmax
9       res.append(word[higest])
10      dist[highest] = -1
11
12  return res
```

Pseudocode 4.7. *Kode untuk mencari 10 kata terdekat dalam ruang spasial Word Embeddinas dari kata tertentu.*

keluar dalam FAQs Schematics sebagai sampel. Penjelasan secara rinci pada uji coba ini akan dijelaskan pada sub-bab 5.1.3.

Implementasi kode pengujian ini terdapat pada Pseudocode 4.7. Fungsi `get_10_nearest` mempunyai sebuah paramerter `word`; `word` adalah kata yang ingin dicari 10 kata terdekat dalam ruang spasialnya. Variabel `vector` pada baris ke-4 menampung sebuah vektor yang merepresentasikan kata `word`. Setelah itu, dihitung nilai *cosine similarity* pada setiap vektor yang ada pada korpus (baris ke-5). Dengan terlebih dahulu membuang nilai *cosine similarity* tertinggi (karena nilainya pasti 1 disebabkan perbandingan dengan kata itu sendiri), dicarilah 10 kata dengan kedekatan tertinggi (baris ke-7 sampai 10). Hasil akhirnya disimpan dalam variabel `res`.

4.5 Implementasi CNN

Pada sub-bab ini akan dijelaskan langkah-langkah dalam implementasi CNN.

4.5.1. Text Preprocessing

Proses *text preprocessing* yang terjadi dengan setiap dokumen pada dataset dapat ditinjau dari Pseudocode 4.8. Fungsi ini menerima satu parameter yang dinamai `text`. Pada baris ke-2, setiap kata yang berpisah dengan spasi dipisahkan menjadi elemen-

```
1   function my_tokenizer(text):
2     txt_space_splitted = text.split(space)
3     remover = regular_expression("[^a-zA-Z-]")
4
5     tokens = inisialisasi array dari token-token
6
7     for word in txt_space_splitted:
8       term = remover.sub('', word)
9       term = term.lower
10      tokens.append(term)
11
12    return tokens
```

Pseudocode 4.8. *Proses text preprocessing setiap dokumen pada dataset.*

elemen pada array. Baris ke-7 sampai ke-10 adalah iterasi pada setiap kata yang menjadi elemen pada array `txt_space_splitted`. Setiap kata dihilangkan unsur non-hurufnya. Kecuali, karakter *dash*. Karena dalam bahasa Indonesia ada kata yang disebut berulang (disambung dengan karakter *dash*) dan dianggap menjadi satu kata. Seperti, kupu-kupu, ubur-ubur, dan lainnya. Setelah itu, setiap kata diubah menjadi *lower case.* Terakhir, kata-kata ini ditambahkan ke dalam *array* yang menyimpan kata-kata yang telah melalui *text preprocessing*.

4.5.2. Sentence Mapping and Averaging

Sebelum setiap dokumen masuk dalam proses *learning* dengan CNN, dokumen-dokumen ini perlu diubah dahulu bentuknya menjadi vektor (*sentence embeddings*). Proses ini disebut dengan *sentence mapping. Sentence* Embeddings adalah vektor yang merepresentasikan sebuah dokumen. Vektor ini tersusun dengan merata-rata setiap Word Embeddings yang merepresentasikan kata yang menyusun dokumen/kalimat.

Implementasi dari proses ini dijelaskan pada Pseudocode 4.9. Fungsi `sentence_to_vec` bertugas untuk mengubah sebuah dokumen menjadi vektor. Pada baris ke-2, dokumen terlebih dahulu dibersihkan dari karakter *break line*. Lalu, dokumen akan

```
1   function sentence_to_vec(text):
2     text = text.replace('\n', '')
3     tokens = my_tokenizer(text)
4
5     begin = True
6     for word in tokens:
7       status, vec = get_word_embd(word)
8       if not status:
9         continue
10      if begin:
11        begin = False
12        feature = vec
13      else:
14        feature += vec
15
16    feature = feature / norm(feature)
17
18    return feature
```

Pseudocode 4.9. *Proses mengubah dokumen menjadi Sentence Embeddings.*

masuk ke dalam *text preprocessing* melalui fungsi `my_tokenizer` (lihat sub-bab 4.2.5). Setiap kata pada dokumen diambil Word Embeddings-nya yang dihasilkan oleh fungsi `get_word_embd` (baris ke-7). Fungsi `get_word_embd` bekerja dengan cara mendapatkan Word Embeddings pada basis data dengan pencarian berdasarkan kata yang masuk. Variabel *boolean* `status` adalah sebuah *flag* untuk menandakan apakah sebuah kata memang ada pada korpus dan mempunyai Word Embeddings. Jika tidak ada, maka langsung lanjut ke kata berikutnya. Baris ke-10 sampai 14 menunjukan proses menambahkan seluruh Word Embeddings dan menampungnya dengan variabel `feature`. Terakhir, baris ke-16, variabel `feature` dibagi dengan normalisasi hasil penjumlahan Word Embeddings.

4.5.3. Memisahkan Dataset

Agar performa CNN dapat dievaluasi, dataset akan dibagi menjadi *data training* dan *data testing*. Proses ini menggunakan *library* Scikit-Learn (Pedregosa et al., 2011). *Library* ini ada pada

```
1   model = Hasil pemodelan CNN
2
3   predictions = model.predict(X_test)
4   predictions = [x.argmax for x in predictions]
5   labels = [x.argmax for x in labels]
6
7   accuracy = metrics.accuracy(labels, predictions)
8   precision = metrics.precision(labels, predictions)
9   recall = metrics.recall(labels, predictions)
10  f1 = metrics.f1_score(labels, predictions)
```

Pseudocode 4.10. *Implementasi prediksi dan evaluasi model CNN.*

kelas model_selection dan fungsinya bernama train_test_split. Pada dataset LAPOR!, *data training* mendapat porsi sebesar 90% dan *data testing* mendapat porsi 10%. Pada dataset FAQs Schematics, pembagian *data training* dan *data testing* masing-masing sebesar 50%.

4.5.4. CNN Learning

Tahap ini adalah tahap inti implementasi CNN. Proses ini melibatkan *feed-forward learn* dan *backpropagation* agar CNN dapat melakukan klasifikasi. Dalam tugas akhir ini, implementasi ini berbeda-beda pada setiap kasus uji coba sehingga penjelasannya secara rinci akan ada pada sub-bab 4.6, 4.7, dan 4.8.

4.5.5. Prediksi dan Evaluasi

Tahap ini adalah tahap terakhir dari implementasi CNN. Evaluasi CNN akan menggunakan akurasi, *precision*, *recall*, dan *F-Measure*. Sebelum dapat menghitung metrik-metrik tesebut, perlu didapatkan hasil prediksi model CNN terhadap *data testing*.

Pada Pseudocode 4.10, dapat dilihat baris ke-3 sampai ke-5 adalah proses mengubah *one hot vector* hasil prediksi model dan *groudtruth* (yang diberi nama variabel `label`) menjadi indeks. Lalu, untuk mendapatkan metriks akurasi, *precision*, *recall*, dan *F-Measure*, digunakan *library* Scikit-Learn (Pedregosa et al., 2011) pada kelas `metrics`.

```
1   X = 2D Matrix of sentences embeddings
2   Y = 2D Matrix of one hot vectors
3
4   model = Sequential()
5   model.add => Conv1D(size=10, kernel_size=3,
6               activation=relu, input_shape=300)
7   model.add => Flatten()
8   model.add => Dropout(p=0.1)
9   model.add => Dense(size=64, activation=sigmoid)
10  model.add => Dropout(p=0.1)
11  model.add => Dense(size=22,activation=softmax)
```

Pseudocode 4.11. *Implementasi kode GloVe dan CNN Tanpa Transfer Learning.*

4.6 Implementasi GloVe dan CNN Tanpa Transfer Learning

Implementasi GloVe dan CNN tanpa Transfer Learning mempunyai langkah-langkah yang sama persis pada sub-bab 4.5. Dataset yang digunakan adalah dataset yang menjadi fokus pada tugas akhir ini, yaitu FAQs Schematics.

Pseudocode 4.11 menjelaskan bagaimana CNN ini diimplementasikan. Variabel `X` dan `Y` masing-masing adalah matriks yang berisi tumpukan Sentence Embeddings setiap dokumen dam tumpukan *one hot vectors* yang merupakan label dari kelas-kelas pada FAQs Schematics. Pada baris ke-5 dan ke-6, terlihat besar kernel yang digunakan pada layer konvolusi adalah sebesar 3. Neuron pada layer konvolusi sebanyak 10. Variabel `input_shape` sebesar 300 adalah ukuran Sentence Embeddings yang menjadi masukan CNN. Neuron pada layer *hidden* sebanyak 64 (baris ke-64). Neuron pada layer output sebanyak 22 sesuai dengan jumlah kelas/label (baris ke-11). Nilai *dropout* yang digunakan masing-masing adalah 0,1 (baris ke-8 dan ke-10).

Pada CNN ini, algoritma *adaptive gradient descent* yang digunakan adalah Adaptive Moment Estimation (Adam) dengan nilai *learning rate* $1x10^{-3}$ dan ε sebesar $1x10^{-8}$. Jumlah *epochs* yang ditetapkan sebanyak 2000 dengan *patience* 10 *epochs.* Jika nilai

```
1   X = 2D Matrix of sentences embeddings
2   Y = 2D Matrix of one hot vectors
3
4   model = Sequential()
5   model.add => Conv1D(size=10, kernel_size=3,
6               activation=relu, input_shape=300)
7   model.add => Flatten()
8   model.add => Dropout(p=0.1)
9   model.add => Dense(size=64, activation=sigmoid)
10  model.add => Dropout(p=0.1)
11  model.add => Dense(size=68,activation=softmax)
```

Pseudocode 4.12. *Implementasi Source Task TS pada Transfer Learning Skenario 1.*

loss sudah tidak turun pada 10 *epochs*, maka proses *learning* CNN akan berhenti.

4.7 Implementasi GloVe, CNN, dan Transfer Learning Skenario 1

Implementasi GloVe, CNN, dan Transfer Learning Skenario 1 melibatkan model CNN yang pernah dilatih dengan data LAPOR!. Model CNN ini kemudian diambil layer konvolusinya untuk digunakan kembali pada CNN yang akan dilatih lagi dengan data FAQs Schematics. Maka dari itu, proses *training* CNN yang dilatih dengan dataset LAPOR! disebut dengan Source Task ($\mathcal{T}_S$) dan proses CNN yang dilatih dengan data FAQs Schematics disebut dengan Target Task ($\mathcal{T}_T$).

Pseudocode 4.12 menunjukan implementasi kode pada $\mathcal{T}_S$. Perbedaan yang ditunjukan oleh implementasi pada Pseudocode 4.11 adalah jumlah neuron pada layer output yang berjumlah 68 sesuai dengan jumlah kelas yang merupakan kategori pada data LAPOR!. Algoritma Adam digunakan kembali dengan *learning rate* $1x10^{-3}$ dan ε $1x10^{-8}$. Jumlah *epochs* yang ditetapkan sebesar 500 dengan *patience* 1 *epochs*. Proses *training* dengan iterasi yang tidak terlalu banyak dilakukan karena jumlah data LAPOR! yang

```
1   ts_model = array of 𝒯_S saved model
2
3   model = Sequential()
4   model.add => ts_model[0]
5   model.add => Flatten()
6   model.add => Dropout(p=0.1)
7   model.add => Dense(size=64, activation=sigmoid)
8   model.add => Dropout(p=0.1)
9   model.add => Dense(size=len(Y[0]),activation=softmax)
10
```

Pseudocode 4.13. *Implementasi Target Task pada Transfer Learning Skenario 1.*

sangat besar. Model terbaik hasil $\mathcal{T}_S$ kemudian disimpan untuk kebutuhan Transfer Learning.

Pseudocode 4.13 menunjukan implementasi kode pada $\mathcal{T}_T$. Variabel `ts_model` adalah variabel yang menyimpan pemodelan CNN hasil dari $\mathcal{T}_S$. Seluruh implementasi CNN ini sama persis seperti pada Pseudocode 4.11. Perbedaannya adalah pada baris ke-4. Layer konvolusi pada CNN ini menggunakan kembali layer konvolusi hasil dari $\mathcal{T}_S$.

Sama seperti implementasi pada sub-bab 4.6, algoritma *adaptive gradient descent* yang digunakan adalah Adaptive Moment Estimation (Adam) dengan nilai *learning rate* $1x10^{-3}$ dan ε sebesar $1x10^{-8}$. Jumlah *epochs* yang ditetapkan sebanyak 2000 dengan *patience* 10 *epochs.*

4.8 Implementasi GloVe, CNN, dan Transfer Learning Skenario 2

Ada 3 CNN yang dilatih pada Transfer Learning Skenario 2. Pertama, CNN dilatih dengan dataset LAPOR! dan menjadi Source Task 1 ($\mathcal{T}_S$-*1*). Kedua, CNN dilatih dengan dataset FAQs Schematics dan menjadi Source Task 2 ($\mathcal{T}_S$-*2*). Ketiga, CNN ini adalah gabungan dari CNN hasil pemodelan $\mathcal{T}_S$-*1* dan $\mathcal{T}_S$-*2* dan hasil gabungannya kemudian dilatih kembali menggunakan dataset

```
1   ts_model_1 = array of T_S-1 saved model
2   ts_model_2 = array of T_S-2 saved model
3
4   model = Sequential()
5   model.add => ts_model_1[0]
6   for layer in ts_model_2[1..n_layer]
7       model.add => layer
```

Pseudocode 4.14. *Implementasi Target Task pada Transfer Learning Skenario 2.*

FAQs Schematics (yang menjadi tujuan utama tugas akhir). Implementasi pada $\mathcal{T}_S$-*1* sama persis dengan yang dilakukan pada Pseudocode 4.12. Implementasi pada $\mathcal{T}_S$-*2* sama persis dengan yang dilakukan pada Pseudocode 4.11.

Setelah didapatkan model terbaik hasil dari $\mathcal{T}_S$-*1* dan $\mathcal{T}_S$-*2*, model CNN baru gabungan kedua model tadi dibentuk. Layer konvolusi diambil dari model hasil dari $\mathcal{T}_S$-*1*. Layer *hidden* dan layer output diambil dari $\mathcal{T}_S$-*1*. Proses ini terlihat dari Pseudocode 4.14. Variabel `ts_model_1` adalah variabel yang menyimpan model hasil dari $\mathcal{T}_S$-*1*. Variabel `ts_model_2` adalah variabel yang menyimpan model dari $\mathcal{T}_S$-*2*. CNN hasil gabungan dari kedua model yang digunakan kembali untuk melatih dataset FAQs Schematics disebut dengan Target Task ($\mathcal{T}_T$). Pada baris ke-5, terlihat layer konvolusi ditambahkan ke arsitektur. Kemudian, seluruh layer setelah konvolusi ditambahkan ke arsitektur (baris ke-5 sampai 7).

Sama seperti implementasi pada sub-bab 4.6, algoritma *adaptive gradient descent* yang digunakan adalah Adaptive Moment Estimation (Adam) dengan nilai *learning rate* $1x10^{-3}$ dan ε sebesar $1x10^{-8}$. Jumlah *epochs* yang ditetapkan sebanyak 2000 dengan *patience* 10 *epochs.*

```
1   dq_dict = file dictionary daftar direct question
2
3   function is_direct_question(chat):
4     for answer, class in dq_dict:
5       if answer == chat:
6         return True, class
7     return False
```

Pseudocode 4.15. *Mengecek apakah sebuah chat adalah direct question.*

4.9 Implementasi Aplikasi *Chatbot*

Pada sub-bab ini, dijelaskan implementasi aplikasi *chatbot* yang dibagi ke dalam tiga sub-bab, yaitu bagaimana *chatbot* menjawab *chat direct questions*, bagaimana sistem menjawab *chat* dengan menggunakan prediksi CNN, dan bagaimana sistem menangani ambiguitas.

4.9.1. Menjawab Direct Questions

Direct Question adalah *chat* yang tidak memerlukan prediksi CNN untuk memberikan jawaban ke pengguna. Jika sebuah *chat* masuk dan diidentifikasi sebagai *direct question* oleh sistem, maka sistem langsung mengirimkan jawaban kepada pengguna (lihat *flowchart* pada Gambar 3.13). Teknik ini digunakan untuk mempermudah sistem ketika pengguna mengirimkan pertanyaan ambigu.

Pada Pseudocode 4.15 dijelaskan fungsi dalam memutuskan apakah sebuah *chat* termasuk *direct question* atau bukan. Variabel `dq_dict` adalah variabel yang berisi struktur data *dictionary* antara daftar *direct question* dengan kelasnya. Ketika sebuah *chat* masuk, fungsi `is_direct_question` akan dipanggil untuk melakukan pengecekan. Perulangan pada baris ke-4 sampai ke-6 akan melakukan iterasi untuk mencocokkan apakah *chat* termasuk *direct question*. Jika ada, fungsi akan mengembalikan nilai `True` dan kode kelas *direct question* tersebut. Jika tidak ada, maka fungsi akan mengembalikan nilai `False`.

```
1   answer_dict = file dictionary daftar jawaban
2
3   function get_answer(class):
4     answer = answer_dict[class]
5     return answer
```

Pseudocode 4.16. *Implementasi sistem dalam mengembalikan jawaban berdasarkan kelas pertanyaan.*

Setelah sebuah *chat* ditentukan kelas pertanyannya, sistem akan mengirim *chat* kepada pengguna dengan jawaban yang sudah menjadi *template*. Daftar jawaban menurut kelasnya dapat dilihat pada Lampiran I.

Pada Pseudocode 4.16 dijelaskan fungsi sebuah sistem dalam menentukan jawaban. Sebuah fungsi `get_answer` akan mengembalikan jawaban berdasarkan data yang ada pada `answer_dict`.

4.9.2. Menjawab Dengan Prediksi CNN dan Menangani Ambiguitas

Bagaimana sistem *chatbot* menjawab *chat* dengan cara memprediksi kelas pertanyaan menggunakan CNN? Cara ini dapat dilihat pada Pseudocode 4.18. Fungsi get_prediction akan menerima sebuat parameter doc yang tidak lain adalah dokumen dari *chat* yang masuk. Dokumen ini terlebih dahulu diubah bentuknya menjadi Sentence Embeddings dengan fungsi sentence_to_vec yang prosesnya dijelaskan pada Pseudocode 4.9 (baris ke-4). Setelah itu, CNN memprediksi kelas dengan mengembalikan nilai probabilitas setiap kelas yang disimpan pada variabel prediction (baris ke-5). Pada variabel prediction, nilai probabilitas kelas pertama menempati indeks pertama. Nilai probabilitas kelas kedua menempati indeks kedua. Begitu seterusnya. Untuk mendeteksi ambiguitas (ketidakyakinan model dalam meprediksi kelas pertanyaan), nilai probabilitas tertinggi dicek besar nilainya. Jika nilai probabilitasnya ada di bawal alpha, maka prediksi ini ambigu. Baris ke-9 sampai ke-13 adalah proses

```
1   model = Model CNN akhir yang digunakan pada chatbot
2
3   function get_prediction(doc):
4     vec = sentence_to_vec(doc)
5     prediction = model.predict(vec)
6     argmax = prediction.argmax
7
8     if prediction[argmax] < alpha:
9       for i in range(3):
10        prediction[argmax] = -1
11        ids.append(argmax)
12        argmax = prediction.argmax
13        return ids
14    else:
15      ids.append(argmax)
16      return ids
```

Pseudocode 4.18. *Proses prediksi jawaban pada sistem chatbot.*

mengembalikan 3 kelas dengan probabilitas tertinggi. Jika nilai probabilitas tertinggi lebih besar dari alpha, maka fungsi ini akan mengembalikan satu kelas itu saja. Nilai alpha yang ditentukan pada tugas akhir ini sebesar 0,7.

Pada Pseudocode 4.17 dijelaskan bagaimana sistem membalas *chat*. Variabel ids adalah variabel hasil dari fungsi get_prediction yang berisi *array* dari ID kelas dengan probabilitas tertinggi. Pada baris ke-2, dicek apakah panjang *array* melebih 1. Jika iya, maka sistem akan mengirim *multiple choice* dari *direct question* tiga kelas yang ada di *array* ids. Jika hanya satu, maka fungsi get_answer akan dipanggil untuk mendapatkan *template* jawaban pada kelas dengan nilai probabilitas tertinggi (baris 5 dan 6).

```
1   function reply(ids):
2     if len(ids) > 1:
3       return multiple_choice
4     else :
5       answer = get_answer(ids[0])
6       return answer
```

Pseudocode 4.17. *Chatbot memberikan respons multiple choice atau jawaban.*

[*Halaman ini sengaja dikosongkan*]

BAB V
UJI COBA DAN EVALUASI

Pada bab ini, akan dijabarkan hasil uji coba beserta evaluasinya. Secara garis besar, uji coba dibagi menjadi tiga bagian besar yang masing-masing menguji kontribusi tiga teori utama yaitu, GloVe, CNN, dan Transfer Learning. GloVe diuji untuk mengetahui apakah Word Embeddings yang dihasilkannya mampu memberikan representasi makna kata yang baik. CNN diuji kemampuannya memprediksi kalimat dengan menggunakan metrik akurasi, *precision, recall*, dan *F-Measure*. Terakhir, metode CNN dengan Transfer Learning akan diuji untuk mengetahui apakah Transfer Learning berhasil meningkatkan performa model.

5.1 GloVe Word Embeddings

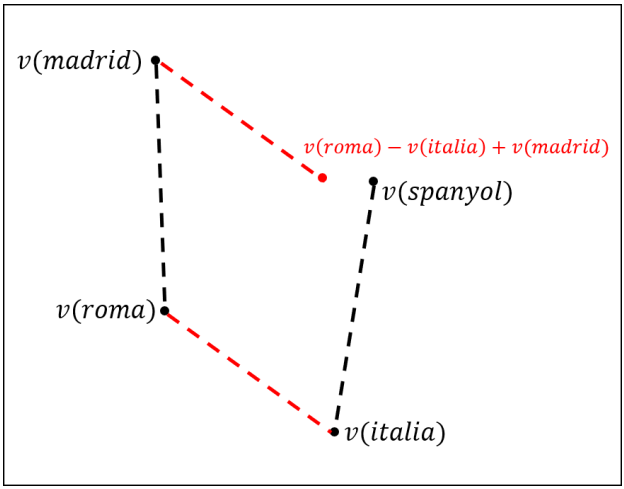
Pada sub-bab ini dijelaskan uji coba dan evaluadi pada Word Embeddings yang dihasilkan oleh algoritma GloVe.

5.1.1 Deskripsi Korpus

Korpus yang digunakan pada tugas akhir ini adalah Wikipedia Bahasa Indonesia dan dokumen-dokumen pada data LAPOR! yang dipublikasi pemerintah Indonesia di *https://lapor.go.id*. Total token yang digunakan adalah sejumlah 83 juta token dengan 2.216.260 kata unik. Kata yang digunakan pada proses *training* adalah kata yang muncul paling sedikit 5 kali dalam korpus. Dengan begitu, ada 331.286 kata unik dalam korpus yang digunakan.

5.1.2 Uji Fungsionalitas Semantik GloVe Word Embeddings

Salah satu cara untuk mengetahui apakah GloVe telah memberikan representasi makna kata berupa vektor dengan baik pada korpus bahasa Indonesia adalah dengan cara memeriksa apakah GloVe dapat mendeteksi makna semantik dari kata. Pada tugas akhir ini, uji semantik dilakukan dengan cara memprediksi pasangan ibukota dan negaranya. Seperti yang terlihat pada



Gambar 5.1. *Ilustrasi salah satu data sintetis untuk uji semantik GloVe Word Embeddings.*

Gambar 5.1, Word Embeddings kata "roma" dan kata "madrid" seharusnya mempunyai jarak yang sama dengan "italia" dan "spanyol". Maka dari itu, operasi vektor $v(roma) - v(italia) + v(madrid)$ seharusnya menghasilkan nilai vektor yang dekat dengan $v(spanyol)$ karena jarak antara ibukota dan negara masing-masing seharusnya mempunyai jarak yang sama (lihat garis putus-putus berwarna merah). Maka dari itu, uji coba dilakukan berdasarkan asumsi ini.

Uji coba semantik Word Embeddings GloVe diawali dengan membuat 506 data sintetis. Satu data sintetis akan terdiri dari $capital1, country1, capital2,$ dan $country2$. Di mana $capital1$ adalah kata yang merupakan ibukota dari $country1$ dan $capital2$ adalah kata yang merupakan ibukota dari $country2$. Pada setiap data sintetis terlebih dahulu didapatkan vektor vec dari hasil operasi $v(capital1) - v(country1) + v(capital2)$. Lalu, dicari kata yang memiliki kemiripan tertinggi dengan vektor vec menggunakan *cosine similarity* pada Word Embeddings yang ada pada seluruh korpus. Dengan menggunakan $country2$ sebagai

Tabel 5.1. *Penggalan hasil uji coba fungsionalitas semantik GloVe Word Embeddings. Hasil selengkapnya terlampir pada Lampiran 2.*

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
1	athena	yunani	baghdad	irak	irak
2	bangkok	thailand	beijing	cina	tiongkok
3	berlin	jerman	hanoi	vietnam	vietnam
4	kairo	mesir	london	inggris	britania
5	canberra	australia	ottawa	kanada	kanada
6	hanoi	vietnam	moskwa	rusia	rusia
...	...	...	...	...	...
506	tokyo	jepang	teheran	iran	iran

groundtruth, hasilnya kemudian dibandingkan. Jika kata yang memiliki kemiripan tertinggi dengan vec adalah $country2$, maka satu kali uji coba dianggap berhasil dan begitupun sebaliknya.

Dengan total data uji sintesis 506 data, hasil uji analogi kata berhasil menggapai tingkat keberhasilan sebesar 86,4% dengan jumlah benar 437 dari 506 data. Penggalan hasil uji coba ini dapat dilihat pada Tabel 5.1 di mana Kota 1 adalah $capital1$, Negara 1 adalah $country1$, Kota 2 adalah $capital2$, dan Negara 2 adalah $country2$ (selengkapnya ada pada Lampiran 2). Jika ditinjau dari hasil pengujian, percobaan no. 2 dan no. 4 tidak menghasilkan kata "cina" dan "inggris". Namun, menghasilkan kata "tiongkok" dan "britania". Walaupun salah, konteks yang ditunjukan tetap mendekati makna dari nama negara-negara tersebut.

GloVe berhasil memberikan representasi makna kata yang cukup akurat dalam korpus Wikipedia Bahasa Indonesia yang sebenarnya tidak terlalu besar jika dibandingkan dengan korpus lain seperti Google News atau Wikipedia Bahasa Inggris yang total tokennya bisa mencapai 6-15 Milyar dengan 3-6 juta kata unik. Karena GloVe mengekstrasi frekuensi kemunculan kata, maka korpus yang lebih besar seharusnya memberikan hasil yang jauh lebih baik.

Keterhubungan antara kata nama negara dan nama ibukotanya biasanya dibahas dalam artikel khusus negara atau ibukota di Wikipedia. Wikipedia Bahasa Inggris mempunyai artikel sejarah

yang jauh lebih lengkap dibandingkan Wikipedia Bahasa Indonesia. Biasanya, dalam artikel sejarah juga sering disebut nama negara dan kota-kota besarnya. Dengan lebih banyaknya jumlah artikel, GloVe juga dapat mempelajari makna kata dengan lebih naik. Kalau ditinjau lebih lanjut lagi, penulisan nama kota kadang tidak konsisten dalam bahasa Indonesia. Misal, ibukota Rusia sering ditulis Moscow, Moskow, atau Moskwa. GloVe menganggap 3 kata ini adalah 3 kata yang berbeda. Dengan penulisan pada korpus yang lebih konsisten, model GloVe bisa mempelajari setiap kata dengan lebih baik lagi.

5.1.3 Uji dan Evaluasi 10 Kata Terdekat

Salah satu alasan digunakannya GloVe pada tugas akhir ini adalah kemampuannya dalam mengatasi beragam kata yang memiliki kata yang mirip. Pada uji coba kali ini, tiga kata dijadikan sampel untuk dilihat apakah kata tersebut mempunyai kedekatan dengan penggunaan kata lain yang muncul di dataset.

Contohnya pada kelas Aturan Seragam di dataset FAQs Schematics (Dataset secara lengkap dapat dilihat di Lampiran 3). Beberapa pertanyaan yang muncul menggunakan kata “seragam”, beberap menggunakan kata “baju”. Dengan menggunakan *cosine similarity*, uji coba kali ini ingin melihat apakah kata “seragam” mempunyai kedekatan dengan kata “baju”. Jika dilihat pada Tabel 5.2, ternyata “seragam” benar mempunyai kedekatan dengan kata “baju” dengan nilai *cosine similarity* 0,56 dan menjadi kata terdekat nomor 4. Lalu ditinjau dari kelas Biaya NLC dan Biaya NPC, bisa dilihat beberapa dokumen menggunakan kata “dibayar”. Beberapa dokumen lain dengan kelas yang sama menggunakan kata “harga”, “biaya”, “rupiah”, dan “uang”. Jika dicari 10 kata terdekat dari kata “dibayar”, maka muncul kedekatan dengan kata “biaya”, “rupiah”, dan “uang” masing-masing dengan nilai *cosine similarity* di atas 0,5. Hanya kata “harga” yang tidak masuk ke 10 besar. Pengambilan sampel uji coba juga dilakukan pada kelas Where When NLC dan Where When NPC (kelas yang berisi pertanyaan kapan dan di mana NLC/NPC dihelat). Beberapa dokumen menggunakan kata “hari”, beberapa di antaranya

Tabel 5.2. *Hasil uji coba 10 kata terdekat pada kata "seragam", "lokasi", dan "kontestan".*

	$X = \text{"}seragam\text{"}$		$X = \text{"}dibayar\text{"}$		$X = \text{"}hari\text{"}$	
	Y	$cos(\theta)$	Y	$cos(\theta)$	Y	$cos(\theta)$
1	pakaian	0,61	membayar	0,68	malam	0,75
2	mengenakan	0,60	dibayarkan	0,65	minggu	0,75
3	periodepenyedia	0,58	gaji	0,59	setiap	0,72
4	**baju**	**0,56**	pembayaran	0,52	bulan	0,69
5	berkapasitas	0,56	**biaya**	**0,52**	senin	0,69
6	kostum	0,56	buruhpekerja	0,52	jumat	0,68
7	berwara	0,55	**rupiah**	**0,52**	sabtu	0,66
8	memakai	0,55	tunai	0,51	sampai	0,66
9	Putripngpakaian	0,53	**uang**	**0,51**	perayaan	0,66
10	celana	0,51	mahal	0,51	**tanggal**	**0,65**

menggunakan kata "tanggal". Jika dilihat 10 kata terdekat, maka muncul kata "tanggal" dengan nilai *cosine similarity* sebesar 0,65.

Dengan hasil uji coba ini, dapat disimpulkan GloVe berhasil memberikan representasi makna yang baik pada setiap kata. Kata-kata yang mempunyai makna yang mirip, memiliki vektor yang saling berdekatan. Tentunya hal ini sangat penting mengingat *chatbot* dalam implementasinya di lapangan, berhadapan dengan berbagai macam pengguna dengan penggunaan ragam kata yang tidak bisa diprediksi. Dengan menggunakan GloVe, permasalahan ini bisa di atasi dengan baik.

5.2 GloVe dan CNN

Uji coba GloVe dan CNN adalah implementasi terhadap dataset FAQs Schematics. Uji coba pada bagian ini bertujuan untuk melihat kemampuan CNN dengan masukan GloVe tanpa Transfer Learning. Berikut adalah penjelasan dalam uji coba ini.

5.2.1 Dataset

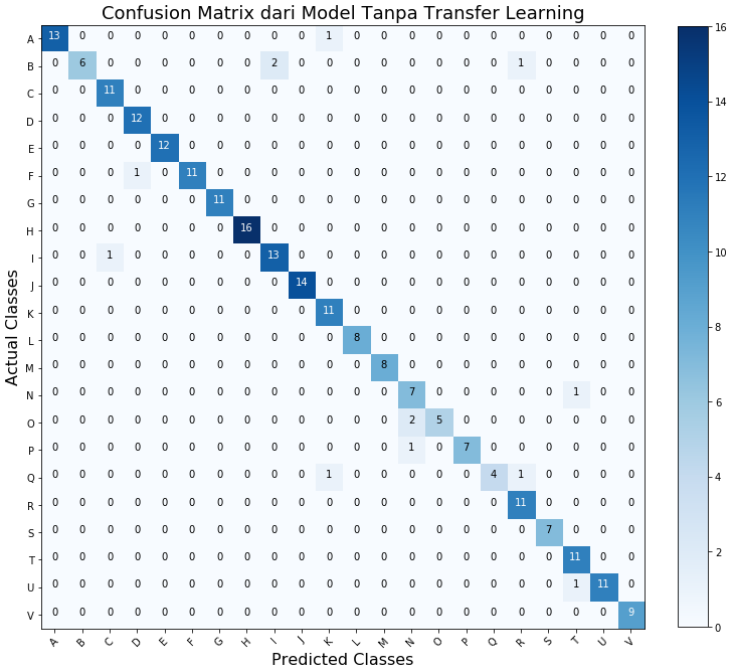
Dataset yang digunakan adalah hasil dari wawancara dengan kepala divisi Humas Schematics 2017. Data ini berupa kalimat-kalimat perintah dan tanya yang sifatnya adalah *frequently asked question* (FAQ) seputar kegiatan Schematics 2017. Perlu diketahui, Schematics 2017 terdiri dari 4 sub-event: *National Logic*

Tabel 5.3. *Contoh beberapa data pada FAQs Schematics. Selengkapnya pada Lampiran 3.*

No.	Kalimat	Kelas
1	mau cek tiket reeva masih ada atau tidak	tiket nst reeva
2	apa yang dilombakan di npc	what is npc
3	kapan sertifikat saya dan kelompok saya datang	sertifikat tidak datang
4	selamat sore	greetings
5	saya ingin ikut npc tapi tidak bisa programming	cannot code yet
6	mendaftar national logic competition	pendaftaran nlc npc
7	berapa harga tiketnya nst	tiket nst reeva
8	mengapa sertifikat saya tidak kunjung datang ya	sertifikat tidak datang
9	apa yang dilombakan di national logic competition	what is nlc
10	saya ingin minta prosedur pendaftaran npc	pendaftaran nlc npc

Competition (NLC), *National Programming Contest*, *National Seminat of Technology* (NST), dan *Revolutionary Entertaiment and Expo With Various Arts* (REEVA). NLC adalah kompetisi logika antar siswa Sekolah Menengah Atas. NPC adalah kompetisi pemrograman antar siswa SMA. NST adalah seminar dan REEVA adalah konser serta expo yang konsumennya adalah masyarakat secara luas.

Dalam pelaksanaannya, peserta ataupun konsumen Schematics secara keseluruhan seringkali menanyakan hal-hal terkait dengan acara-acara ini seperti: baju apa yang digunakan saat kompetisi, berapa harga tiket seminar, di mana lokasi acara tertentu diadakan, dan lain-lain. Maka dari itu, dikumpulkanlah data-data kalimat ini dari hasil wawancara dengan kepala divisi Humas Schematics 2017, kemudian setiap kalimat ini dilabeli berdasarkan kategori jawabannya. Jumlah data yang berhasil dikumpulkan adalah 462 data dengan 22 kelas jawaban. Beberapa contoh kalimat beserta labelnya dapat dilihat pada Tabel 5.3.

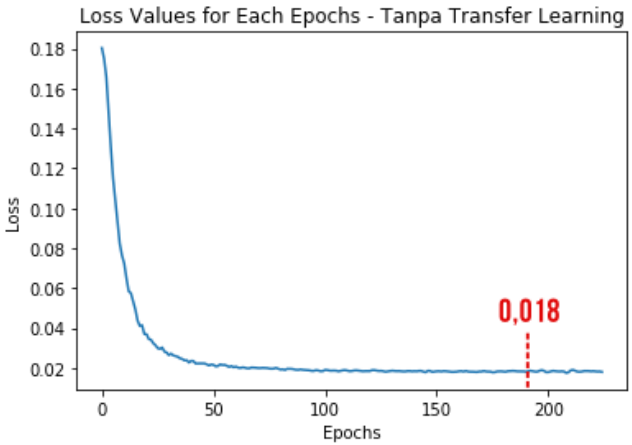


Gambar 5.2. *Confusion matrix dari model GloVe dan CNN tanpa transfer learning.*

5.2.2 Uji dan Evaluasi Kemampuan Prediksi Model

Setelah dataset disiapkan, kemudian dataset dibagi menjadi *train set* dan *test set*. *Train set* adalah bagian data yang menjadi *training* bagi CNN. Sedangkan *test set* adalah bagian data yang akan digunakan untuk mendapatkan nilai *accuracy*, *precision*, *recall*, dan *F-Measure*. Porsi pembagian dataset ini masing-masing adalah 50% dengan alasan agar *train set* dikondisikan relatif kecil karena asumsi awal bahwa perusahaan/organisasi yang ingin membuat *chatbot* tidak akan mempunyai banyak data. Jadi, total *train set* dan *test set* masing-masing adalah 231.

Setelah dilatih dengan total maksimal 2000 *epochs* dan *early stopper* pada 10 *epochs*, model mencapai nilai *loss* terbaik pada angka 0,018 (Gambar 5.3). Model ini tidak bisa lebih baik lagi



Gambar 5.3. *Nilai loss per epochs pada model tanpa transfer learning*

karena nilai loss mulai naik secara konstan tidak lama setelah mencapai nilai terendahnya. Lalu, model diukur kembali dengan cara menghitung hasil prediksi dengan *test set* dengan hasil *accuracy*, *precision*, *recall*, dan *F-Measure* masing-masing sebesar 94,4%; 95,2%; 94,4%; dan 94,3% (Tabel 5.6). Gambar 5.2 menunjukan *confusion matrix* hasil dari evaluasi model ini dan Tabel 5.4 adalah keterangan labelnya.

Dapat disimpulkan kombinasi CNN dan GloVe ternyata memberikan hasil yang sangat baik (semua nilai metrik menunjukan nilai diatas 94%). Dengan didahului hasil GloVe yang sudah berhasil memberikan representasi makna kata dengan bagus, CNN menjadi jauh lebih mudah dalam melakukan klasfikasi.

5.3 GloVe, CNN, dan Transfer Learning

Tujuan dari tahap uji coba ini adalah berusaha menaikkan kembali nilai *accuracy*, *precision*, *recall*, dan *F-Measure* dengan teknik Transfer Learning. Transfer Learning akan melibatkan $\mathcal{D}_S$ sebagai data pada $\mathcal{T}_S$ yang hasil akhir modelnya ditansfer untuk digunakan kembali oleh $\mathcal{T}_T$.

***Tabel 5.4**. Tabel keterangan kelas pada confusion matrix.*

Label	Kelas
A	Greetings
B	What is Schematics
C	What is NLC
D	What is NPC
E	Benefit NLC
F	Benefit NPC
G	Soal NLC
H	Cannot Code Yet
I	Where When NLC
J	Where When NPC
K	Pendaftaran NLC NPC
L	Timeline NLC
M	Timeline NPC
N	Biaya NLC
O	Biaya NPC
P	Belum Verifikasi
Q	Pindah Region NLC
R	Jumlah Kelompok
S	Sertifikat Tidak Datang
T	Tiket NST Reeva
U	Pengumuman NLC NPC
V	Aturan Seragam

5.3.2 Dataset

Dataset akhir yang akan digunakan pada uji coba pada tahap ini sama persis dengan dataset yang telah dijelaskan pada sub-bab 5.2.1. Perbedaannya dengan uji coba pada sub-bab 5.2.2 adalah CNN yang akan dilatih dengan $\mathcal{D}_{\mathcal{T}}$ akan dilatih dulu dengan $\mathcal{D}_{\mathcal{S}}$. Kemudian, pada tahap uji coba ini akan dilakukan eksperimen arsitektur seperti yang sudah dijelaskan pada sub-bab 3.4.4 dan 3.4.5.

LAPOR! adalah web portal yang dimiliki pemerintah Indonesia yang bertujuan untuk memberikan fasilitas kepada masyarakat Indonesia melaporkan berbagai hal kepada pemerintah agar pemerintah bisa mengambil tindakan langsung. Data-data ini sudah

Tabel 5.5. *Beberapa contoh data pada dataset LAPOR!.*

No.	Pelapor	Isi Laporan	Kategori
1	628214771xxxx	Pak Menteri Kesehatan. Tolong perhatikan kami…	Bidang Kesejahteraan Rakyat
2	628574613xxxx	Kemenkumham, standardisasi minimal pornografi &…	Teknologi Informasi dan Komunikasi
3	628139296xxxx	Menkes RI yang terhormat tolong inspeksi ke R…	Kesehatan
4	628215460xxxx	Selemat pagi, saya Andrianto, dokter PTT di pe…	Kesehatan
5	62816114xxxx	Yth. Bpk.Kapolda Kalimantan Tengah. Penegak h…	Lingkungan Hidup dan Penanggulangan Bencana
6	628529934xxxx	Yth. Menpan, kapan tenaga honorer kategori 2 a…	Pendidikan
7	628137169xxxx	Mohon dicek ulang tentang seleksi penerimaan B…	Reformasi Birokrasi dan Tata Kelola
8	628138824xxxx	Yth. Bapak Kemenkeu, kami Pegawai Negeri Sipil…	Perdagangan, Perindustrian, Iklim…
9	628135968xxxx	Di Jombang, Jawa Timur, terdapat penipuan menj…	Reformasi Birokrasi dan Tata Kelola
10	628533836xxxx	Mohon ditegur dokter-dokter yang bertugas di K…	Reformasi Birokrasi dan Tata Kelola

terkategori ketika dipublikasi. Maka dari itu, CNN dilatih dengan menggunakan laporan-laporan ini dengan kategori laporan sebagai labelnya. Total data yang berhasil dikumpulkan dalam kurun waktu tahun 2011-2016 ada 87.764 total data.

Sebelum dilakukan klasifikasi, data ini dibagi menjadi *train set* (90%) dan *test set* (10%) sehingga total *train set* menjadi 78.988 dan total *test set* menjadi 8.776. Pembagian yang sangat besar pada *train set* bertujuan untuk memberikan data *training* yang lebih besar pada CNN. Seperti yang sudah dijelaskan pada publikasi-publikasi tentang Transfer Learning (Pan et al., 2010; Yosinski et al., 2014; Semwal et al., 2018), besarnya data $\mathcal{D}_S$ sangat berpengaruh pada keberhasilan transfer *knowledge*. Contoh penggalan dataset LAPOR! ada pada Tabel 5.5.

5.3.3 Uji dan Evaluasi Kemampuan Prediksi Model

Setelah proses $\mathcal{T}_S$ dilakukan dengan menggunakan $\mathcal{D}_S$, model hasil dari $\mathcal{T}_S$ digunakan kembali dalam proses transfer *knowledge*

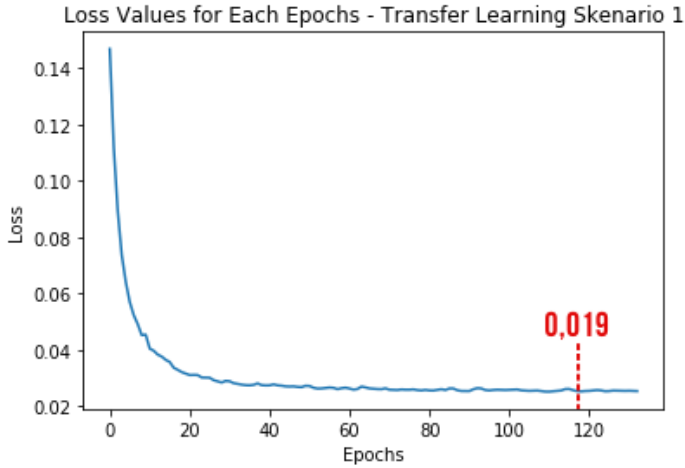
Tabel 5.6. *Perbandingan $\mathcal{T}_T$ sebelum Transfer Learning, Transfer Learning Skenario (1), dan Transfer Learning Skenario (2).*

Skenario	*Loss*	*Accuracy*	*Precision*	*Recall*	*F1*
Sebelum Transfer Learning	0,018	94,4%	95,2%	94,4%	94,3%
Transfer Learning Skenario (1)	0,019	92,2%	93,2%	92,2%	92,2%
Transfer Learning Skenario (2)	**0,013**	**95,7%**	**96,4%**	**95,7%**	**95,6%**

(yang sebenarnya adalah menggunakan kembali layer-layer yang sudah terboboti oleh hasil *training* dengan $\mathcal{D}_S$). Dalam hal ini, seperti yang sudah dijelaskan pada sub-bab 3.4.4 dan 3.4.5, ada 2 skenario dalam proses Transfer Learning: Transfer Learning Skenario 1, yaitu dengan hanya menggunakan layer `Conv1D` hasil dari $\mathcal{T}_S$ lalu menginisialisasi ulang layer-layer lain dan Transfer Learning Skenario 2, yaitu dengan menggunakan layer `Conv1D` hasil dari $\mathcal{T}_S$*-1* lalu menggunakan layer-layer lain hasil dari $\mathcal{T}_S$*-2* (hasil CNN dijelaskan pada sub-bab 4.8).

Agar perbandingannya menjadi adil, semua skenario Transfer Learning sama-sama dilatih dengan menggunakan *epochs* sebanyak 2000 dengan *early stopper* sebanyak 10 *epochs*. Nilai *loss* terbaik yang didapatkan adalah 0,019 pada Transfer Learning Skenario 1 dan 0,013 pada Transfer Learning Skenario 2 (lihat grafik pada Gambar 5.4 dan Gambar 5.5). Kedua model ini kemudian diukur kemampuan prediksinya dengan menggunakan *test set* yang hasilnya bisa dilihat pada Tabel 5.6.

Hasil dari uji coba ini menunjukan hasil yang menarik. Transfer Learning Skenario 1 tidak berhasil mengungguli model tanpa transfer learning. Inisialisasi ulang layer-layer lain dengan menggunakan `Conv1D` hasil dari $\mathcal{T}_S$ ternyata memberikan hasil yang lebih buruk. Kemungkinan sebabnya adalah proses *learning* yang tidak imbang antara `Conv1D` dengann layer-layer lain. `Conv1D` sudah melalui proses *learning* yang panjang, sedangkan layer-layer lain dalam keadaan belum pernah melalui proses *learning* sama

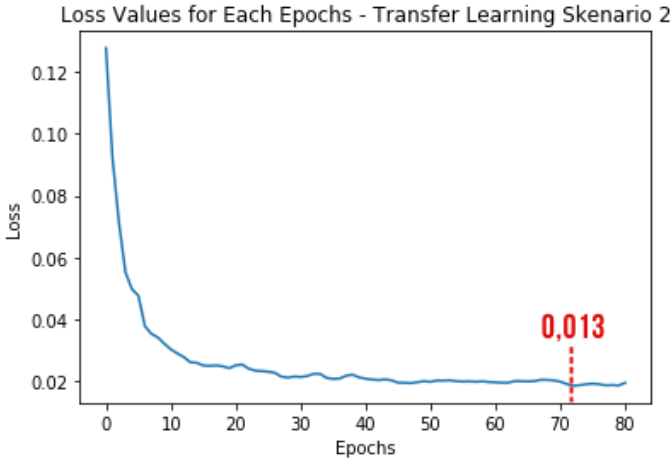


Gambar 5.4. *Nilai loss per epochs pada Skenario Transfer Learning 1*

sekali. Hasil inilah yang juga menjadi inspirasi Transfer Learning Skenario 2, yaitu menggabungkan kedua model hasil dari $\mathcal{T}_s$*-1* dan $\mathcal{T}_s$*-2*. Dengan menggunakan layer-layer yang telah melalui proses learning sebelumnya, dengan menempatkan layer `Conv1D` yang melakukan tugas umum dan menempatkan layer lainnya yang terlah dilatih dengan tugas spesifik, model ini berhasil memberikan peningkatan pada performa. Ini dikarenakan layer `Conv1D` berhasil melakukan tugasnya dalam mengidentifikasi fitur penting dalam Sentence Embeddings (pekerjaan umum dalam setiap layer konvolusi) dan keberhasilan layer-layer lain setelahnya yang berhasil meningkatkan kemampuan klasifikasi karena sebelumnya telah dilatih melakukan klasifkasi dengan dataset yang sama. Hasil kedua skenario Transfer Learning ini juga dapat dievaluasi melalui *confusion matrix* pada Gambar 5.7 dan Gambar 5.8 dan keterangan label pada Tabel 5.4.

5.3.4 Uji dan Evaluasi Kecepatan Belajar Model

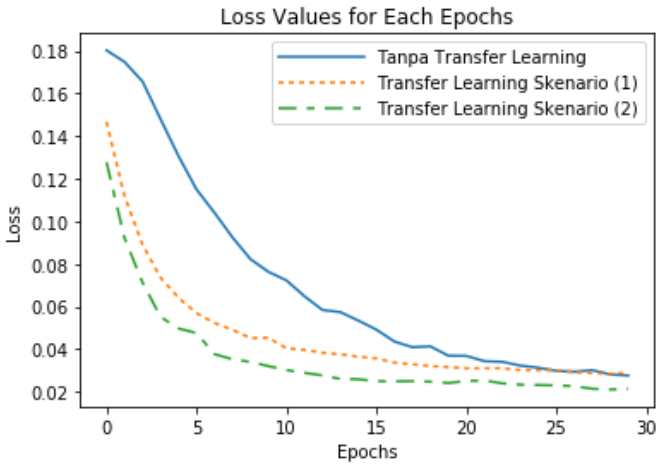
Dengan asumsi CNN telah melalui proses *learning* sebelumnya, nilai-nilai bobot neuron-neuron pada CNN seharusnya sudah



Gambar 5.5. *Nilai loss per epochs pada Skenario Transfer Learning 2*

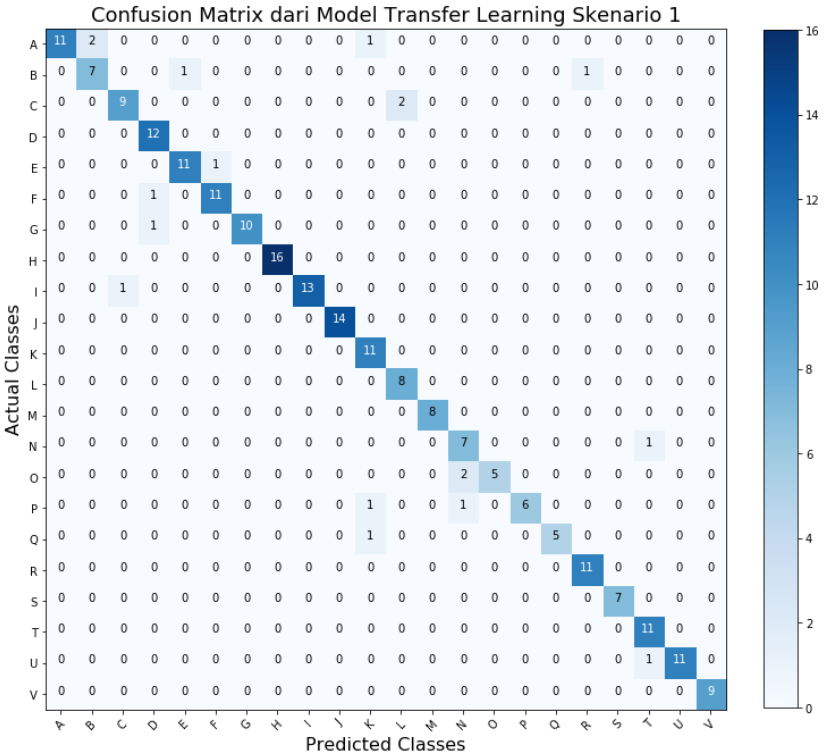
mendekati nilai optimal. Maka, wajarlah jika muncul hipotesis bahwa CNN mendapat transfer seharusnya dapat melakukan proses *training* dengan *epochs* yang lebih sedikit (lebih cepat). Oleh karena itu, uji coba pada tahap ini meninjau dari seberapa cepat nilai *loss* berkurang pada setiap *epochs*.

Hasil uji coba pada tahap ini bisa dilihat dari Gambar 5.6. Jika grafik pada Gambar 5.4 ditinjau, model CNN tanpa Transfer Learning melalui proses *learning* dengan lebih lambat. Ini dikarekanan bobot pada neuron-neuron CNN diinisialisasi secara acak sehingga membutuhkan waktu yang lebih panjang untuk meminimalisasi nilai *loss.* Sedangkan CNN dengan Transfer Learning melakukan *training* jauh lebih cepat. Terlihat dengan nilai *loss* yang berkurang lebih besar pada setiap *epochs*. Ini dikarenakan nilai bobot neuron-neuron sudah mendekati optimum karena telah dilatih data yang mirip (pada Transfer Learning Skenario 1) dan data yang sama persis (pada Transfer Learning Skenario 2). Karena bobot pada neuron-neuron model Transfer Learning Skenario 2 hampir seluruhnya menggunakan hasil dari

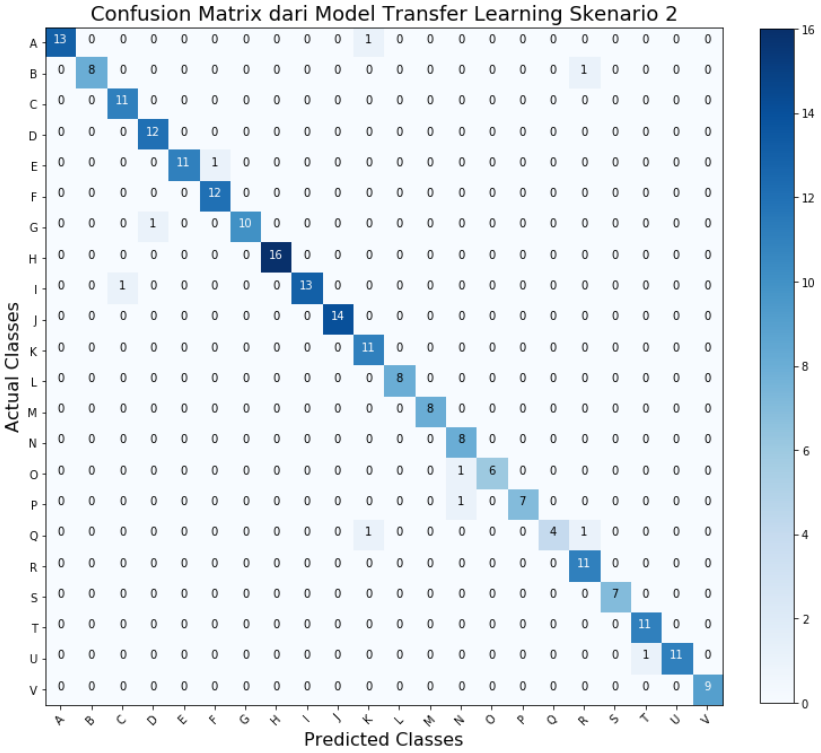


Gambar 5.6. *Kecepatan belajar tiap model ditinjau dari nilai loss per epochs*

learning sebelumnya, tidak hanya lebih cepat, tetapi juga memulai dengan nilai *loss* yang lebih rendah dibandingkan model yang lain.



Gambar 5.7. *Confusion matrix dari model Transfer Learning Skenario 1*



***Gambar 5.8**. Confusion matrix dari model Transfer Learning Skenario 2.*

BAB VI
KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan kesimpulan dan saran pengembangan dari tugas akhir ini.

6.1 Kesimpulan

Dari berbagai uji coba yang telah dilakukan, terdapat empat poin kesimpulan dari tugas akhir ini, yaitu:

1. Algoritma GloVe berhasil menghasilkan hasil Word Embeddings yang memberikan representasi makna kata yang sangat baik. Terbukti dengan hasil dari uji analogi kata pada kasus nama negara dan ibukotanya yang menggapai tingkat kebenaran hingga 86,4%. Selain itu, GloVe juga memberikan hasil yang sangat memuaskan ketika diminta untuk memberikan kata-kata dengan makna terdekat dari kata kunci yang diberikan.
2. GloVe dan CNN terbukti berhasil melakukan klasifikasi kombinasi-kombinasi kalimat dan dapat memprediksi kombinasi kalimat lain dengan akurat. Hal ini didasari dengan hasil uji coba yang tingkat *accuracy*, *precision*, *recall*, dan *F-Measure*-nya masing-masing mencapai 94,4%; 95,2%; 94,4%; dan 94,3%.
3. Kombinasi GloVe, CNN, dan Transfer Learning berhasil mengatasi data yang sedikit. Dengan hanya bermodalkan 231 data latih, kombinasi model hasil dari perkawinan tiga algoritma ini berhasil mencapai angka *accuracy*, *precision*, *recall*, dan *F-Measure* mencapai 95,7%; 96,4%; 95,7%; dan 95,6% pada prediksi data dengan jumlah yang sama. Maka dari itu, dapat disimpulkan sistem *chatbot* yang dikembangkan pada tugas akhir ini mampu memahami makna kata dan memprediksi jawaban pada berbagai macam kombinasi kalimat dengan data latih awal yang relatif sedikit.
4. Berdasarkan hasil dari uji coba pada bab 5.3.4, Transfer Learning juga memberikan keuntungan lain. Dengan meninjau

kecepatan iterasi (*epochs*) setiap model dalam meminimalisasi nilai *loss*, proses *learning* pada model dengan Transfer Learning jauh lebih cepat dibandingkan dengan CNN tanpa Transfer Learning.

6.2 Saran

Berikut ini adalah saran untuk pengembangan penelitian selanjutnya.

1. Perlu diuji coba pengaruh besarnya data korpus pada performa GloVe. Karena GloVe mempelajari makna kata dari rasio kemungkinan kemunculan kata, maka frekuensi kemunculan kata yang lebih tinggi seharusnya memberikan GloVe informasi yang lebih.
2. Semwal et al. (2018) menekankan bahwa $\mathcal{D}_S$ yang dipilih harus semirip mungkin dengan $\mathcal{D}_T$ pada kasus klasfikasi teks. Jika tidak, maka proses Transfer Learning akan gagal. Oleh karena itu, diperlukan penelitan bagaimana caranya secara empiris dapat mengukur kemiripan dataset sehingga proses pemilihan Ds tidak hanya berdasarkan asumsi saja.
3. Mengombinasikan Word Embeddings GloVe dengan TF-IDF. TF-IDF pada dasarnya adalah metode yang sangat bagus dalam menentukan ukuran relevansi sebuah kata dalam dokumen. Kata yang paling unik dalam sebuah dokumen, seharusnya menjadi representasi menonjol dalam dokumen tersebut. Oleh karena itu, dapat diasumsikan bahwa mengalikan nilai TF-IDFT sebuah kata dengan Word Embeddings-nya akan menghasilkan representasi yang lebih akurat sehingga memudahkan CNN dalam melakukan klasifikasi.
4. Menambahkan *stopwords* dalam *preprocessing* pada setiap dokumen yang ingin diklasifikasikan. Stopwords adalah kata yang sangat sering muncul dalam sebuah bahasa sehingga relevansinya terhadap representasi dokumen sangat rendah. Dengan menghilangkan *stopwords* pada dokumen, dapat diasumsikan hanya kata yang membunyai relevansi tinggilah yang kemudian akan diklasifikasikan.

DAFTAR PUSTAKA

Agus Zainal Arifin dan Ari Novan Setiono. 2002. Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia Dengan Algoritma Single Pass Clustering. *Prosiding Seminar on Intellegent Technology and Its Application (SITIA)*.

Alex Krizhevsky, Ilya Sutskever, dan Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25.*

Baotian Hu, Zhengdong Lu, Hang Li, dan Qingcai Chen. 2015. Convolutional Neural Network Architectures for Matching Natural Language Senteces. *NIPS 2015*.

Bayan Abu Shawar dan Eric Atwell. 2007. Chatbots: Are they Really Useful. *LDV-Forum*, 22:29-49.

Diederik P. Kingma dan Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representation (ICLR).*

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, dan Édouard Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825-2830.

Honglak Lee, Roger Grosse, Rajesh Ranganath, dan Andrew Y. Ng. 2009. *ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning*, hal. 609-616.

Jack Kiefer dan Jacob Wolfowitz. 1952. Stochastic Estimation of the Maximum of a Regression Function. *Function. Ann. Math. Statist. 23*, no. 23, hal. 462-466.

Jason Yosinski, Jeff Clune, Yoshua Bengio, dan Hod Lipson. 2014. How Transferable are Features in Deep Neural Networks. *Advances in Neural Information Processing Systems 27*, hal. 320-3328.

Jeffrey Pennington, Richard Socher, dan Christoper D. Manning. 2014. GloVe: Global Vectors for Word Representation. *In Proceedings Empirical Methods in Natural Language Processing (EMNLP)*, hal. 1532-1543.

John A. Bullinaria dan Joseph P. Levy. 2007. Extracting Semantic Representations From Word Co-Occurrence Statistics: A Computational Study. *Behavior Research Methods*, 39(3):510-526.

John Duchi, Elad Hazan, dan Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research 12*, hal. 2121-2159.

Joseph Weizenbaum. 1966. ELIZA – A Computer Program for The Study of Natural Language Communication Between Man and Machine. *Communications of the ACM*, 10(8):474-480.

Joyce Chai, Jimmy Lin, Wlodek Zadrozny, Yiming Ye, Margo Stys-Budzikowska, Veronika Horvath, Nana Kambhatla, dan Catherine Wolf. 2001. The Role of A Natural Language Conversational Interface in Online Sales: A Case Study. *International Journal of Speech Technology*, 4:285:295.

Juan Ramos. 2003. Using TF-IDF to Determine Word Relevance in Document Queries.

Kevin Lund and Curt Burgess. 1996. Producing High-Dimensional Semantic Spaces From Lexical Co-Occurrence, *Behavior Research Methods, Instrumentation, and Computers*, 28:203:208.

Nal Kalchbrenner, Edward Grefenstette, dan Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. *In Proceedings of ACL 2014.*

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, dan Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15*, hal. 1929-1958.

Putu Praba Santika, Agus Zainal Arifin, dan Diana Purwitasari. 2015. Pembentukan Thesaurus Yang Sensitif Terdahap Tingkat Polaritas Review Pada Cross-Domain Sentiment Classification. *Jurnal Inspiration.*

Quoc V. Le, Alexandre Karpenko, Kiquan Ngiam, dan Andrew Y. Ng. 2011. ICA with Reconstruction Cost for Efficient Overcomplete Feature Leaning. *Advances in Neural Information Processing Systems 24 (NIPS 2011).*

Richard S. Wallace. 2003. The Elements of AIML Style. A.L.I.C.E. Artificial Intellegence Foundation, Inc.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, dan Pavel Kuksa. 2011. Natural Language Processing (Almost) From Scratch. *Journal of Machine Learning Research 12*, hal. 2493-2537.

Scott W. Yih, Xiaodong He, dan Chris Meek. 2014. Semantic Parsing for Single-Relation Question Answering. *In Proceedings of ACL 2014.*

Sinno Jialin Pan dan Qian Yiang. 2010. A Survey on Transfer Learning. *IEEE Transactions On Knowledge and Data Engineering*, vol. 22, no. 10.

Thomas K. Landauer, Peter W. Foltz, dan Darrel Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.

Tom Kenter, Alexey Borisov, dan Maarten de Rijke. 2016. Siamese CBOW: Optimization Word Embeddings for Sentence Representations. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, hal. 941-951.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, dan Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *In ICLR Workshop Papers.*

Tushar Semwal, Gaurav Mathur, Promod Yenigalla, dan Shiravshankar B. Nair. 2018. A Practitioners' Guide to Transfer Learning for Text Classification using Convolutional Neural Networks. *Accepted paper in SDM.*

Wlodek Zadrozny, Margo Stys-Budzikowska, Joyce Chai, Nana Kambhatla, S. Levesque, dan N. Nicolov. 2000. Natural Language Dialogue for Personalized Interaction. *Communication of the ACM*, 43(8):116-120.

Yelong Shen, Xiaodong He, Jiafeng Gao, Li Deng, dan Gregoire Mesnil. 2014. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. *In Proceedings of WWW 2014.*

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *In Proceedings Empirical Methods in Natural Language Processing (EMNLP).*

LAMPIRAN

Lampiran 1 – Daftar Jawaban Chatbot

Kelas	Jawaban
Greetings	Halo. Namaku Miranda Rahma. Kamu bisa panggil aku Mira. Aku adalah chatbot yang bertugas untuk menjawab pertanyaanmu tentang Schematics 2018. Salam kenal ya!
What is Schematics	Schematics 2018 terdiri dari 4 event besar. National Logic Competition (NLC), National Programming Contest (NPC), National Seminat of Technology (NST), dan REEVA. Dipersembahkan oleh HMTC Informatika ITS.
What is NLC	National Logic Competition (NLC) adalah kompetisi logika terbesar se-Indonesia yang diadakan oleh HMTC Informatika ITS. NLC diadakan dengan format tim (maksimal 3 orang) siswa SMA/SMK/sederajat dan berasal dari sekolah yang sama.
What is NPC	National Programming Contest (NPC) adalah kompetisi pemrograman tingkat nasional. Kompetisi ini diadakan oleh HMTC Informatis ITS. NPC diadakan dengan format individu/perorangan siswa SMA/SMK/sederajat. Bahasa pemrograman yang digunakan adalah C/C++ dan Pascal.
Benefit NLC	Untuk ikut NLC, kamu tidak perlu menghafal materi. Cukup andalkan logika aja. Selain bisa dapat sertifikasi nasional, belajar untuk Tes Potensi Akademik (TPA), dan bisa dapat gratis biaya transportasi (untuk finalis), pemenang

Kelas	Jawaban
	juara 1 bisa langsung diterima di Informatika ITS lho!
Benefit NPC	Keuntungan ikut NPC kamu bisa dapat pembelajaran pemrograman yang tidak kamu dapat di SMA. Selain bisa dapat sertifikat nasional, pemenang NPC bisa langsung diterima di Informatika ITS lho!
Soal NLC	Contoh soal NLC: 1. (X-A)(X-B)(X-C).....(X-Z)= ... 2. Pada sebuah balapan lari, anda dibalap oleh pelari yang ada di posisi ke-2. Berada di mana posisi anda sekarang? 3. M, V, B, M, J, S, U, ... (Apa huruf berikutnya?)
Cannot Code Yet	Tenang aja. NPC menyediakan modul tutorial pemrograman C/C++ dan tutorial online buat kamu yang belum bisa coding dan pengen belajar lho! Yuk, langsung daftar di schematics.its.ac.id dan dapatkan modul tutorialnya.
Where When NLC	NLC akan diadakan: MINGGU, 1 OKTOBER 2018 7.00 s/d selesai Di Regionalnya masing-masing
Where When NPC	NLC: 1. Via offline, hubungi:\nRaihan (0812345678) Hero (0887654321) 2. Via online, daftar melalui website + bayar melalui ATM *Pendaftaran ditutup tanggal 7 September 2018\nBiaya: Rp150.000/tim NPC: 1. Via offline, hubungi:

Kelas	Jawaban
	Raihan (0812345678) Hero (0887654321) 2. Via online, daftar melalui website + bayar melalui ATM *Pendaftaran ditutup tanggal 7 September 2018 Biaya: Rp50.000/tim
Pendaftaran NLC NPC	NLC: 1. Via offline, hubungi:\nRaihan (0812345678) Hero (0887654321) 2. Via online, daftar melalui website + bayar melalui ATM *Pendaftaran ditutup tanggal 7 September 2018\nBiaya: Rp150.000/tim NPC: 1. Via offline, hubungi: Raihan (0812345678) Hero (0887654321) 2. Via online, daftar melalui website + bayar melalui ATM *Pendaftaran ditutup tanggal 7 September 2018 Biaya: Rp50.000/tim
Timeline NLC	*Kirim Gambar Timeline NLC*
Timeline NPC	*Kirim Gambar Timeline NPC*
Biaya NLC	Biaya pendaftaran NLC adalah Rp150.000/tim ya! Semangat!
Biaya NPC	Biaya pendaftaran NPC adalah Rp50.000/orang ya! Semangat!
Belum Verifikasi	Waktu verfikasi maksimal adalah seminggu setelah tanggal pembayaran ya. Harap sabar menunggu. Terima kasih! :)
Pindah Region NLC	Bagi kamu yang ingin pindah region NLC, kalian bisa ikuti prosedur ini ya.

Kelas	Jawaban
	kirim gambar sop pindah region
Jumlah Kelompok	Jumlah anggota kelompok NLC maksimal 3 orang. Boleh sendiri, boleh berdua, tetapi tidak boleh lebih dari tiga ya. Untuk NPC, lombanya individu alias satu orang saja.
Sertifikat Tidak Datang	Untuk kamu yang sertifikatnya belum turun, bisa melakukan pengajuan kembali permintaan sertifikat melalui prosedur ini ya. Terima kasih. *kirim gambar sop permintaan sertifikat*
Tiket NST Reeva	Tiker NST dan REEVA masih ada nih. Miliki segera sebelum kehabisan. Langsung kunjungi link berikut! http://schematics.its.ac.id
Pengumuman NLC NPC	*Kirim gambar timeline pasca penyisihan*
Aturan Seragam	Pakaian yang dikenakan saat lomba adalah seragam putih abu-abu sekolah masing-masing ya. Semangat!
Tidak Ada Pilihan	Mohon maaf aku masih belajar :(Mungkin kamu bisa tanya ulang dengan kalimat yang berbeda. :)

Lampiran 2 – Hasil Uji Fungsionalias Semantik GloVe Word Embeddings

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
1	athena	yunani	baghdad	irak	irak
2	athena	yunani	bangkok	thailand	thailand
3	athena	yunani	beijing	cina	tiongkok
4	athena	yunani	berlin	jerman	jerman
5	athena	yunani	bern	swiss	swiss
6	athena	yunani	kairo	mesir	mesir
7	athena	yunani	canberra	australia	australia
8	athena	yunani	hanoi	vietnam	vietnam
9	athena	yunani	havana	kuba	kuba
10	athena	yunani	helsinki	finlandia	finlandia
11	athena		islamabad	pakistan	pakistan
12	athena	yunani	kabul	afganistan	afganistan
13	athena	yunani	london	inggris	inggris
14	athena	yunani	madrid	spanyol	real
15	athena	yunani	moskwa	rusia	rusia
16	athena	yunani	oslo	norwegia	norwegia
17	athena	yunani	ottawa	kanada	kanada
18	athena	yunani	paris	perancis	perancis
19	athena	yunani	roma	italia	katolik
20	athena	yunani	stockholm	swedia	swedia
21	athena	yunani	teheran	iran	iran
22	athena	yunani	tokyo	jepang	jepang
23	baghdad	irak	bangkok	thailand	thailand

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
24	baghdad	irak	beijing	cina	tiongkok
25	baghdad	irak	berlin	jerman	jerman
26	baghdad	irak	bern	swiss	swiss
27	baghdad	irak	kairo	mesir	mesir
28	baghdad	irak	canberra	australia	australia
29	baghdad	irak	hanoi	vietnam	vietnam
30	baghdad	irak	havana	kuba	kuba
31	baghdad	irak	helsinki	finlandia	finlandia
32	baghdad	irak	islamabad	pakistan	pakistan
33	baghdad	irak	kabul	afganistan	afganistan
34	baghdad	irak	london	inggris	britania
35	baghdad	irak	madrid	spanyol	real
36	baghdad	irak	moskwa	rusia	rusia
37	baghdad	irak	oslo	norwegia	norwegia
38	baghdad	irak	ottawa	kanada	kanada
39	baghdad	irak	paris	perancis	perancis
40	baghdad	irak	roma	italia	katolik
41	baghdad	irak	stockholm	swedia	swedia
42	baghdad	irak	teheran	iran	iran
43	baghdad	irak	tokyo	jepang	jepang
44	baghdad	irak	athena	yunani	yunani
45	bangkok	thailand	beijing	cina	tiongkok
46	bangkok	thailand	berlin	jerman	jerman
47	bangkok	thailand	bern	swiss	swiss
48	bangkok	thailand	kairo	mesir	mesir

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
49	bangkok	thailand	canberra	australia	australia
50	bangkok	thailand	hanoi	vietnam	vietnam
51	bangkok	thailand	havana	kuba	kuba
52	bangkok	thailand	helsinki	finlandia	finlandia
53	bangkok	thailand	islamabad	pakistan	pakistan
54	bangkok	thailand	kabul	afganistan	afganistan
55	bangkok	thailand	london	inggris	britania
56	bangkok	thailand	madrid	spanyol	real
57	bangkok	thailand	moskwa	rusia	rusia
58	bangkok	thailand	oslo	norwegia	norwegia
59	bangkok	thailand	ottawa	kanada	kanada
60	bangkok	thailand	paris	perancis	perancis
61	bangkok	thailand	roma	italia	italia
62	bangkok	thailand	stockholm	swedia	swedia
63	bangkok	thailand	teheran	iran	iran
64	bangkok	thailand	tokyo	jepang	jepang
65	bangkok	thailand	athena	yunani	yunani
66	bangkok	thailand	baghdad	irak	irak
67	beijing	cina	berlin	jerman	jerman
68	beijing	cina	bern	swiss	swiss
69	beijing	cina	kairo	mesir	mesir
70	beijing	cina	canberra	australia	australia
71	beijing	cina	hanoi	vietnam	vietnam
72	beijing	cina	havana	kuba	kuba
73	beijing	cina	helsinki	finlandia	finlandia

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
74	beijing	cina	islamabad	pakistan	pakistan
75	beijing	cina	kabul	afganistan	afganistan
76	beijing	cina	london	inggris	britania
77	beijing	cina	madrid	spanyol	real
78	beijing	cina	moskwa	rusia	rusia
79	beijing	cina	oslo	norwegia	norwegia
80	beijing	cina	ottawa	kanada	kanada
81	beijing	cina	paris	perancis	perancis
82	beijing	cina	roma	italia	katolik
83	beijing	cina	stockholm	swedia	swedia
84	beijing	cina	teheran	iran	schenkel
85	beijing	cina	tokyo	jepang	jepang
86	beijing	cina	athena	yunani	yunani
87	beijing	cina	baghdad	irak	muslim
88	beijing	cina	bangkok	thailand	thailand
89	berlin	jerman	bern	swiss	swiss
90	berlin	jerman	kairo	mesir	mesir
91	berlin	jerman	canberra	australia	australia
92	berlin	jerman	hanoi	vietnam	vietnam
93	berlin	jerman	havana	kuba	kuba
94	berlin	jerman	helsinki	finlandia	finlandia
95	berlin	jerman	islamabad	pakistan	pakistan
96	berlin	jerman	kabul	afganistan	afganistan
97	berlin	jerman	london	inggris	inggris
98	berlin	jerman	madrid	spanyol	spanyol

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
99	berlin	jerman	moskwa	rusia	rusia
100	berlin	jerman	oslo	norwegia	norwegia
101	berlin	jerman	ottawa	kanada	kanada
102	berlin	jerman	paris	perancis	perancis
103	berlin	jerman	roma	italia	italia
104	berlin	jerman	stockholm	swedia	swedia
105	berlin	jerman	teheran	iran	iran
106	berlin	jerman	tokyo	jepang	jepang
107	berlin	jerman	athena	yunani	yunani
108	berlin	jerman	baghdad	irak	irak
109	berlin	jerman	bangkok	thailand	thailand
110	berlin	jerman	beijing	cina	tiongkok
111	bern	swiss	kairo	mesir	mesir
112	bern	swiss	canberra	australia	australia
113	bern	swiss	hanoi	vietnam	vietnam
114	bern	swiss	havana	kuba	kuba
115	bern	swiss	helsinki	finlandia	finlandia
116	bern	swiss	islamabad	pakistan	pakistan
117	bern	swiss	kabul	afganistan	afganistan
118	bern	swiss	london	inggris	britania
119	bern	swiss	madrid	spanyol	spanyol
120	bern	swiss	moskwa	rusia	rusia
121	bern	swiss	oslo	norwegia	norwegia
122	bern	swiss	ottawa	kanada	kanada
123	bern	swiss	paris	perancis	perancis

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
124	bern	swiss	roma	italia	italia
125	bern	swiss	stockholm	swedia	swedia
126	bern	swiss	teheran	iran	iran
127	bern	swiss	tokyo	jepang	jepang
128	bern	swiss	athena	yunani	yunani
129	bern	swiss	baghdad	irak	irak
130	bern	swiss	bangkok	thailand	thailand
131	bern	swiss	beijing	cina	tiongkok
132	bern	swiss	berlin	jerman	jerman
133	kairo	mesir	canberra	australia	australian
134	kairo	mesir	hanoi	vietnam	vietnam
135	kairo	mesir	havana	kuba	kuba
136	kairo	mesir	helsinki	finlandia	finlandia
137	kairo	mesir	islamabad	pakistan	pakistan
138	kairo	mesir	kabul	afganistan	afganistan
139	kairo	mesir	london	inggris	britania
140	kairo	mesir	madrid	spanyol	real
141	kairo	mesir	moskwa	rusia	rusia
142	kairo	mesir	oslo	norwegia	norwegia
143	kairo	mesir	ottawa	kanada	kanada
144	kairo	mesir	paris	perancis	perancis
145	kairo	mesir	roma	italia	romawi
146	kairo	mesir	stockholm	swedia	swedia
147	kairo	mesir	teheran	iran	iran
148	kairo	mesir	tokyo	jepang	jepang

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
149	kairo	mesir	athena	yunani	yunani
150	kairo	mesir	baghdad	irak	irak
151	kairo	mesir	bangkok	thailand	thailand
152	kairo	mesir	beijing	cina	tiongkok
153	kairo	mesir	berlin	jerman	jerman
154	kairo	mesir	bern	swiss	swiss
155	canberra	australia	hanoi	vietnam	vietnam
156	canberra	australia	havana	kuba	kuba
157	canberra	australia	helsinki	finlandia	finlandia
158	canberra	australia	islamabad	pakistan	pakistan
159	canberra	australia	kabul	afganistan	afganistan
160	canberra	australia	london	inggris	britania
161	canberra	australia	madrid	spanyol	real
162	canberra	australia	moskwa	rusia	rusia
163	canberra	australia	oslo	norwegia	norwegia
164	canberra	australia	ottawa	kanada	kanada
165	canberra	australia	paris	perancis	perancis
166	canberra	australia	roma	italia	italia
167	canberra	australia	stockholm	swedia	swedia
168	canberra	australia	teheran	iran	iran
169	canberra	australia	tokyo	jepang	jepang
170	canberra	australia	athena	yunani	yunani
171	canberra	australia	baghdad	irak	irak
172	canberra	australia	bangkok	thailand	thailand
173	canberra	australia	beijing	cina	tiongkok

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
174	canberra	australia	berlin	jerman	jerman
175	canberra	australia	bern	swiss	swiss
176	canberra	australia	kairo	mesir	mesir
177	hanoi	vietnam	havana	kuba	kuba
178	hanoi	vietnam	helsinki	finlandia	finlandia
179	hanoi	vietnam	islamabad	pakistan	pakistan
180	hanoi	vietnam	kabul	afganistan	afganistan
181	hanoi	vietnam	london	inggris	britania
182	hanoi	vietnam	madrid	spanyol	real
183	hanoi	vietnam	moskwa	rusia	rusia
184	hanoi	vietnam	oslo	norwegia	norwegia
185	hanoi	vietnam	ottawa	kanada	kanada
186	hanoi	vietnam	paris	perancis	perancis
187	hanoi	vietnam	roma	italia	katolik
188	hanoi	vietnam	stockholm	swedia	swedia
189	hanoi	vietnam	teheran	iran	iran
190	hanoi	vietnam	tokyo	jepang	jepang
191	hanoi	vietnam	athena	yunani	yunani
192	hanoi	vietnam	baghdad	irak	irak
193	hanoi	vietnam	bangkok	thailand	thailand
194	hanoi	vietnam	beijing	cina	tiongkok
195	hanoi	vietnam	berlin	jerman	jerman
196	hanoi	vietnam	bern	swiss	swiss
197	hanoi	vietnam	kairo	mesir	mesir
198	hanoi	vietnam	canberra	australia	australia

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
199	havana	kuba	helsinki	finlandia	finlandia
200	havana	kuba	islamabad	pakistan	pakistan
201	havana	kuba	kabul	afganistan	afganistan
202	havana	kuba	london	inggris	britania
203	havana	kuba	madrid	spanyol	real
204	havana	kuba	moskwa	rusia	rusia
205	havana	kuba	oslo	norwegia	norwegia
206	havana	kuba	ottawa	kanada	kanada
207	havana	kuba	paris	perancis	perancis
208	havana	kuba	roma	italia	katolik
209	havana	kuba	stockholm	swedia	swedia
210	havana	kuba	teheran	iran	iran
211	havana	kuba	tokyo	jepang	jepang
212	havana	kuba	athena	yunani	yunani
213	havana	kuba	baghdad	irak	irak
214	havana	kuba	bangkok	thailand	thailand
215	havana	kuba	beijing	cina	tiongkok
216	havana	kuba	berlin	jerman	jerman
217	havana	kuba	bern	swiss	swiss
218	havana	kuba	kairo	mesir	mesir
219	havana	kuba	canberra	australia	australia
220	havana	kuba	hanoi	vietnam	vietnam
221	helsinki	finlandia	islamabad	pakistan	pakistan
222	helsinki	finlandia	kabul	afganistan	afganistan
223	helsinki	finlandia	london	inggris	britania

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
224	helsinki	finlandia	madrid	spanyol	spanyol
225	helsinki	finlandia	moskwa	rusia	rusia
226	helsinki	finlandia	oslo	norwegia	norwegia
227	helsinki	finlandia	ottawa	kanada	kanada
228	helsinki	finlandia	paris	perancis	perancis
229	helsinki	finlandia	roma	italia	italia
230	helsinki	finlandia	stockholm	swedia	swedia
231	helsinki	finlandia	teheran	iran	iran
232	helsinki	finlandia	tokyo	jepang	jepang
233	helsinki	finlandia	athena	yunani	yunani
234	helsinki	finlandia	baghdad	irak	irak
235	helsinki	finlandia	bangkok	thailand	thailand
236	helsinki	finlandia	beijing	cina	tiongkok
237	helsinki	finlandia	berlin	jerman	jerman
238	helsinki	finlandia	bern	swiss	swiss
239	helsinki	finlandia	kairo	mesir	mesir
240	helsinki	finlandia	canberra	australia	australia
241	helsinki	finlandia	hanoi	vietnam	vietnam
242	helsinki	finlandia	havana	kuba	kuba
243	islamabad	pakistan	kabul	afganistan	afganistan
244	islamabad	pakistan	london	inggris	britania
245	islamabad	pakistan	madrid	spanyol	spanyol
246	islamabad	pakistan	moskwa	rusia	rusia
247	islamabad	pakistan	oslo	norwegia	norwegia
248	islamabad	pakistan	ottawa	kanada	kanada

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
249	islamabad	pakistan	paris	perancis	perancis
250	islamabad	pakistan	roma	italia	italia
251	islamabad	pakistan	stockholm	swedia	swedia
252	islamabad	pakistan	teheran	iran	iran
253	islamabad	pakistan	tokyo	jepang	jepang
254	islamabad	pakistan	athena	yunani	yunani
255	islamabad	pakistan	baghdad	irak	irak
256	islamabad	pakistan	bangkok	thailand	thailand
257	islamabad	pakistan	beijing	cina	tiongkok
258	islamabad	pakistan	berlin	jerman	jerman
259	islamabad	pakistan	bern	swiss	swiss
260	islamabad	pakistan	kairo	mesir	mesir
261	islamabad	pakistan	canberra	australia	australia
262	islamabad	pakistan	hanoi	vietnam	vietnam
263	islamabad	pakistan	havana	kuba	kuba
264	islamabad	pakistan	helsinki	finlandia	finlandia
265	kabul	afganistan	london	inggris	britania
266	kabul	afganistan	madrid	spanyol	real
267	kabul	afganistan	moskwa	rusia	rusia
268	kabul	afganistan	oslo	norwegia	norwegia
269	kabul	afganistan	ottawa	kanada	kanada
270	kabul	afganistan	paris	perancis	perancis
271	kabul	afganistan	roma	italia	katolik
272	kabul	afganistan	stockholm	swedia	swedia
273	kabul	afganistan	teheran	iran	iran

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
274	kabul	afganistan	tokyo	jepang	jepang
275	kabul	afganistan	athena	yunani	yunani
276	kabul	afganistan	baghdad	irak	irak
277	kabul	afganistan	bangkok	thailand	thailand
278	kabul	afganistan	beijing	cina	tiongkok
279	kabul	afganistan	berlin	jerman	jerman
280	kabul	afganistan	bern	swiss	swiss
281	kabul	afganistan	kairo	mesir	mesir
282	kabul	afganistan	canberra	australia	australia
283	kabul	afganistan	hanoi	vietnam	vietnam
284	kabul	afganistan	havana	kuba	kuba
285	kabul	afganistan	helsinki	finlandia	finlandia
286	kabul	afganistan	islamabad	pakistan	pakistan
287	london	inggris	madrid	spanyol	spanyol
288	london	inggris	moskwa	rusia	rusia
289	london	inggris	oslo	norwegia	norwegia
290	london	inggris	ottawa	kanada	kanada
291	london	inggris	paris	perancis	perancis
292	london	inggris	roma	italia	katolik
293	london	inggris	stockholm	swedia	swedia
294	london	inggris	teheran	iran	iran
295	london	inggris	tokyo	jepang	jepang
296	london	inggris	athena	yunani	yunani
297	london	inggris	baghdad	irak	irak
298	london	inggris	bangkok	thailand	thailand

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
299	london	inggris	beijing	cina	tiongkok
300	london	inggris	berlin	jerman	jerman
301	london	inggris	bern	swiss	wankdorf
302	london	inggris	kairo	mesir	mesir
303	london	inggris	canberra	australia	australia
304	london	inggris	hanoi	vietnam	vietnam
305	london	inggris	havana	kuba	kuba
306	london	inggris	helsinki	finlandia	finlandia
307	london	inggris	islamabad	pakistan	pakistan
308	london	inggris	kabul	afganistan	afganistan
309	madrid	spanyol	moskwa	rusia	rusia
310	madrid	spanyol	oslo	norwegia	norwegia
311	madrid	spanyol	ottawa	kanada	kanada
312	madrid	spanyol	paris	perancis	perancis
313	madrid	spanyol	roma	italia	katolik
314	madrid	spanyol	stockholm	swedia	swedia
315	madrid	spanyol	teheran	iran	iran
316	madrid	spanyol	tokyo	jepang	jepang
317	madrid	spanyol	athena	yunani	yunani
318	madrid	spanyol	baghdad	irak	irak
319	madrid	spanyol	bangkok	thailand	thailand
320	madrid	spanyol	beijing	cina	tiongkok
321	madrid	spanyol	berlin	jerman	jerman
322	madrid	spanyol	bern	swiss	swiss
323	madrid	spanyol	kairo	mesir	mesir

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
324	madrid	spanyol	canberra	australia	australia
325	madrid	spanyol	hanoi	vietnam	vietnam
326	madrid	spanyol	havana	kuba	kuba
327	madrid	spanyol	helsinki	finlandia	finlandia
328	madrid	spanyol	islamabad	pakistan	pakistan
329	madrid	spanyol	kabul	afganistan	afganistan
330	madrid	spanyol	london	inggris	britania
331	moskwa	rusia	oslo	norwegia	norwegia
332	moskwa	rusia	ottawa	kanada	kanada
333	moskwa	rusia	paris	perancis	perancis
334	moskwa	rusia	roma	italia	katolik
335	moskwa	rusia	stockholm	swedia	swedia
336	moskwa	rusia	teheran	iran	iran
337	moskwa	rusia	tokyo	jepang	jepang
338	moskwa	rusia	athena	yunani	yunani
339	moskwa	rusia	baghdad	irak	irak
340	moskwa	rusia	bangkok	thailand	thailand
341	moskwa	rusia	beijing	cina	tiongkok
342	moskwa	rusia	berlin	jerman	jerman
343	moskwa	rusia	bern	swiss	swiss
344	moskwa	rusia	kairo	mesir	mesir
345	moskwa	rusia	canberra	australia	australia
346	moskwa	rusia	hanoi	vietnam	vietnam
347	moskwa	rusia	havana	kuba	kuba
348	moskwa	rusia	helsinki	finlandia	finlandia

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
349	moskwa	rusia	islamabad	pakistan	pakistan
350	moskwa	rusia	kabul	afganistan	afganistan
351	moskwa	rusia	london	inggris	britania
352	moskwa	rusia	madrid	spanyol	spanyol
353	oslo	norwegia	ottawa	kanada	kanada
354	oslo	norwegia	paris	perancis	perancis
355	oslo	norwegia	roma	italia	italia
356	oslo	norwegia	stockholm	swedia	swedia
357	oslo	norwegia	teheran	iran	iran
358	oslo	norwegia	tokyo	jepang	jepang
359	oslo	norwegia	athena	yunani	yunani
360	oslo	norwegia	baghdad	irak	irak
361	oslo	norwegia	bangkok	thailand	thailand
362	oslo	norwegia	beijing	cina	tiongkok
363	oslo	norwegia	berlin	jerman	jerman
364	oslo	norwegia	bern	swiss	swiss
365	oslo	norwegia	kairo	mesir	mesir
366	oslo	norwegia	canberra	australia	australia
367	oslo	norwegia	hanoi	vietnam	vietnam
368	oslo	norwegia	havana	kuba	kuba
369	oslo	norwegia	helsinki	finlandia	finlandia
370	oslo	norwegia	islamabad	pakistan	pakistan
371	oslo	norwegia	kabul	afganistan	afganistan
372	oslo	norwegia	london	inggris	britania
373	oslo	norwegia	madrid	spanyol	spanyol

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
374	oslo	norwegia	moskwa	rusia	rusia
375	ottawa	kanada	paris	perancis	perancis
376	ottawa	kanada	roma	italia	italia
377	ottawa	kanada	stockholm	swedia	swedia
378	ottawa	kanada	teheran	iran	iran
379	ottawa	kanada	tokyo	jepang	jepang
380	ottawa	kanada	athena	yunani	yunani
381	ottawa	kanada	baghdad	irak	irak
382	ottawa	kanada	bangkok	thailand	thailand
383	ottawa	kanada	beijing	cina	tiongkok
384	ottawa	kanada	berlin	jerman	jerman
385	ottawa	kanada	bern	swiss	swiss
386	ottawa	kanada	kairo	mesir	mesir
387	ottawa	kanada	canberra	australia	australia
388	ottawa	kanada	hanoi	vietnam	vietnam
389	ottawa	kanada	havana	kuba	kuba
390	ottawa	kanada	helsinki	finlandia	finlandia
391	ottawa	kanada	islamabad	pakistan	pakistan
392	ottawa	kanada	kabul	afganistan	afganistan
393	ottawa	kanada	london	inggris	britania
394	ottawa	kanada	madrid	spanyol	spanyol
395	ottawa	kanada	moskwa	rusia	rusia
396	ottawa	kanada	oslo	norwegia	norwegia
397	paris	perancis	roma	italia	katolik
398	paris	perancis	stockholm	swedia	swedia

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
399	paris	perancis	teheran	iran	iran
400	paris	perancis	tokyo	jepang	jepang
401	paris	perancis	athena	yunani	yunani
402	paris	perancis	baghdad	irak	irak
403	paris	perancis	bangkok	thailand	thailand
404	paris	perancis	beijing	cina	tiongkok
405	paris	perancis	berlin	jerman	jerman
406	paris	perancis	bern	swiss	swiss
407	paris	perancis	kairo	mesir	mesir
408	paris	perancis	canberra	australia	australia
409	paris	perancis	hanoi	vietnam	vietnam
410	paris	perancis	havana	kuba	kuba
411	paris	perancis	helsinki	finlandia	finlandia
412	paris	perancis	islamabad	pakistan	pakistan
413	paris	perancis	kabul	afganistan	afganistan
414	paris	perancis	london	inggris	inggris
415	paris	perancis	madrid	spanyol	spanyol
416	paris	perancis	moskwa	rusia	rusia
417	paris	perancis	oslo	norwegia	norwegia
418	paris	perancis	ottawa	kanada	kanada
419	roma	italia	stockholm	swedia	swedia
420	roma	italia	teheran	iran	iran
421	roma	italia	tokyo	jepang	jepang
422	roma	italia	athena	yunani	yunani
423	roma	italia	baghdad	irak	irak

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
424	roma	italia	bangkok	thailand	thailand
425	roma	italia	beijing	cina	tiongkok
426	roma	italia	berlin	jerman	jerman
427	roma	italia	bern	swiss	moutier
428	roma	italia	kairo	mesir	mesir
429	roma	italia	canberra	australia	australian
430	roma	italia	hanoi	vietnam	vietnam
431	roma	italia	havana	kuba	kuba
432	roma	italia	helsinki	finlandia	finlandia
433	roma	italia	islamabad	pakistan	pakistan
434	roma	italia	kabul	afganistan	afganistan
435	roma	italia	london	inggris	britania
436	roma	italia	madrid	spanyol	real
437	roma	italia	moskwa	rusia	rusia
438	roma	italia	oslo	norwegia	norwegia
439	roma	italia	ottawa	kanada	kanada
440	roma	italia	paris	perancis	perancis
441	stockholm	swedia	teheran	iran	iran
442	stockholm	swedia	tokyo	jepang	jepang
443	stockholm	swedia	athena	yunani	yunani
444	stockholm	swedia	baghdad	irak	irak
445	stockholm	swedia	bangkok	thailand	thailand
446	stockholm	swedia	beijing	cina	tiongkok
447	stockholm	swedia	berlin	jerman	jerman
448	stockholm	swedia	bern	swiss	swiss

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
449	stockholm	swedia	kairo	mesir	mesir
450	stockholm	swedia	canberra	australia	australia
451	stockholm	swedia	hanoi	vietnam	vietnam
452	stockholm	swedia	havana	kuba	kuba
453	stockholm	swedia	helsinki	finlandia	finlandia
454	stockholm	swedia	islamabad	pakistan	pakistan
455	stockholm	swedia	kabul	afganistan	afganistan
456	stockholm	swedia	london	inggris	britania
457	stockholm	swedia	madrid	spanyol	spanyol
458	stockholm	swedia	moskwa	rusia	rusia
459	stockholm	swedia	oslo	norwegia	norwegia
460	stockholm	swedia	ottawa	kanada	kanada
461	stockholm	swedia	paris	perancis	perancis
462	stockholm	swedia	roma	italia	italia
463	teheran	iran	tokyo	jepang	jepang
464	teheran	iran	athena	yunani	yunani
465	teheran	iran	baghdad	irak	irak
466	teheran	iran	bangkok	thailand	thailand
467	teheran	iran	beijing	cina	tiongkok
468	teheran	iran	berlin	jerman	jerman
469	teheran	iran	bern	swiss	swiss
470	teheran	iran	kairo	mesir	mesir
471	teheran	iran	canberra	australia	australia
472	teheran	iran	hanoi	vietnam	vietnam
473	teheran	iran	havana	kuba	kuba

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
474	teheran	iran	helsinki	finlandia	finlandia
475	teheran	iran	islamabad	pakistan	pakistan
476	teheran	iran	kabul	afganistan	afganistan
477	teheran	iran	london	inggris	britania
478	teheran	iran	madrid	spanyol	spanyol
479	teheran	iran	moskwa	rusia	rusia
480	teheran	iran	oslo	norwegia	norwegia
481	teheran	iran	ottawa	kanada	kanada
482	teheran	iran	paris	perancis	perancis
483	teheran	iran	roma	italia	italia
484	teheran	iran	stockholm	swedia	swedia
485	tokyo	jepang	athena	yunani	yunani
486	tokyo	jepang	baghdad	irak	irak
487	tokyo	jepang	bangkok	thailand	thailand
488	tokyo	jepang	beijing	cina	tiongkok
489	tokyo	jepang	berlin	jerman	jerman
490	tokyo	jepang	bern	swiss	wankdorf
491	tokyo	jepang	kairo	mesir	mesir
492	tokyo	jepang	canberra	australia	australia
493	tokyo	jepang	hanoi	vietnam	vietnam
494	tokyo	jepang	havana	kuba	kuba
495	tokyo	jepang	helsinki	finlandia	finlandia
496	tokyo	jepang	islamabad	pakistan	pakistan
497	tokyo	jepang	kabul	afganistan	afganistan
498	tokyo	jepang	london	inggris	inggris

No.	Kota 1	Negara 1	Kota 2	Negara 2 (Groundtruth)	Hasil
499	tokyo	jepang	madrid	spanyol	spanyol
500	tokyo	jepang	moskwa	rusia	rusia
501	tokyo	jepang	oslo	norwegia	norwegia
502	tokyo	jepang	ottawa	kanada	kanada
503	tokyo	jepang	paris	perancis	perancis
504	tokyo	jepang	roma	italia	katolik
505	tokyo	jepang	stockholm	swedia	swedia
506	tokyo	jepang	teheran	iran	iran

Lampiran 3 – Dataset FAQs Schematics

No.	Dokumen	Kelas
1	halo	greetings
2	selamat pagi	greetings
3	selamat siang	greetings
4	selamat sore	greetings
5	selamat malam	greetings
6	permisi	greetings
7	apakah ini contact person schematics	greetings
8	apakah ini kontak person schematics	greetings
9	halo selamat pagi	greetings
10	halo selamat sore	greetings
11	halo selamat siang	greetings
12	halo selamat malam	greetings
13	perkenalkan nama saya saya ingin menanyakan sesuatu	greetings
14	perkenalkan saya dari tim dan sekolah ingin menanyakan sesuatu	greetings
15	siang	greetings
16	pagi	greetings
17	sore	greetings
18	malam	greetings
19	permisi apakah benar ini dari schematics	greetings
20	halo apa kabar	greetings
21	bagaimana kabar anda	greetings
22	apa benar ini schematics	greetings
23	permisi apa benar ini schematics	greetings

No.	Dokumen	Kelas
24	apakah benar ini narahubung	greetings
25	saya ingin bertanya sesuatu	greetings
26	ada yang ingin saya tanyakan	greetings
27	apa kabar	greetings
28	siapa anda	greetings
29	apa sih schematics itu	what is schematics
30	terdiri dari apa saja kegiatan schematics	what is schematics
31	apa saja event yang ada di schematics	what is schematics
32	coba ceritakan apa saja isinya schematics	what is schematics
33	schematics	what is schematics
34	siapa yang mengadakan event schematics	what is schematics
35	schematics diadakan oleh siapa	what is schematics
36	mengapa schematics diadakan	what is schematics
37	event apa saja yang ada di schematics	what is schematics
38	event nya schematics	what is schematics
39	kegiatannya schematics apa saja sih	what is schematics
40	kegiatannya schematics apa aja sih	what is schematics
41	apa itu schematics	what is schematics
42	coba ceritakan tentang schematics	what is schematics
43	mohon jelaskan pada saya tentang schematics	what is schematics
44	jelaskan kepada saya apa itu schematics dong	what is schematics
45	ceritakan schematics dong	what is schematics
46	event schematics	what is schematics
47	sebenarnya acara apa schematics ini	what is schematics

No.	Dokumen	Kelas
48	nlc	what is nlc
49	national logic competition	what is nlc
50	national logic competitions	what is nlc
51	apa itu nlc	what is nlc
52	apa itu national logic competition	what is nlc
53	definisikan nlc	what is nlc
54	definisikan national logic competition	what is nlc
55	apa sih nlc itu	what is nlc
56	apa sih national logic competition itu	what is nlc
57	lomba apa sih nlc itu	what is nlc
58	lomba apa sih national logic competition itu	what is nlc
59	apa yang dilombakan di nlc	what is nlc
60	apa yang dilombakan di national logic competition	what is nlc
61	lomba yang seperti apa sih nlc itu	what is nlc
62	lomba yang seperti apa sih national logic competition itu	what is nlc
63	jelaskan pada saya nlc	what is nlc
64	jelaskan pada saya national logic competiion	what is nlc
65	mohon terangkan nlc	what is nlc
66	mohon terangkan national logic competition	what is nlc
67	bisa kah anda jelaskan pada saya tentang nlc	what is nlc
68	bisa kah anda jelaskan pada saya tentang national logic competition	what is nlc

No.	Dokumen	Kelas
69	apa yang dilombakan di nlc	what is nlc
70	apa yang dilombakan di national logic competition	what is nlc
71	npc	what is npc
72	national programming contest	what is npc
73	national programming contests	what is npc
74	apa itu npc	what is npc
75	apa itu national programming contest	what is npc
76	definisikan npc	what is npc
77	definisikan national programming contest	what is npc
78	apa sih npc itu	what is npc
79	apa sih national programming contest itu	what is npc
80	lomba apa sih npc itu	what is npc
81	lomba apa sih national programming contest itu	what is npc
82	apa yang dilombakan di npc	what is npc
83	apa yang dilombakan di national programming contest	what is npc
84	lomba yang seperti apa sih npc itu	what is npc
85	lomba yang seperti apa sih national programming contest itu	what is npc
86	jelaskan pada saya npc	what is npc
87	jelaskan pada saya national programming contest	what is npc
88	mohon terangkan npc	what is npc
89	mohon terangkan national programming contest	what is npc

No.	Dokumen	Kelas
90	bisa kah anda jelaskan pada saya tentang npc	what is npc
91	bisa kah anda jelaskan pada saya tentang national programming contest	what is npc
92	apa yang dilombakan di npc	what is npc
93	apa yang dilombakan di national programming contest	what is npc
94	kenapa harus ikut nlc	benefit nlc
95	kenapa harus ikut national logic competition	benefit nlc
96	mengapa harus ikut nlc	benefit nlc
97	mengapa harus ikut national logic competition	benefit nlc
98	apa yang menarik dari nlc	benefit nlc
99	apa yang menarik dari national logic competition	benefit nlc
100	apa keuntungan dari nlc	benefit nlc
101	apa keuntungan dari national logic competition	benefit nlc
102	apa untungnya ikut nlc	benefit nlc
103	apa untungnya ikut national logic competition	benefit nlc
104	mengapa nlc menguntungkan bagi saya	benefit nlc
105	mengapa national logic competition menguntungkan bagi saya	benefit nlc
106	apa yang bisa saya dapatkan dari nlc	benefit nlc
107	apa yang bisa saya dapatkan dari national logic competition	benefit nlc

No.	Dokumen	Kelas
108	jelaskan pada saya apa keuntungan mengikuti nlc	benefit nlc
109	jelaskan pada saya apa keuntungan mengikuti national logic competition	benefit nlc
110	jelaskan mengapa saya harus ikut nlc	benefit nlc
111	jelaskan mengapa saya harus ikut national logic competition	benefit nlc
112	terangkan pada saya benefit mengikuti nlc	benefit nlc
113	terangkan pada saya benefit mengikuti national logic competition	benefit nlc
114	keuntungan nlc	benefit nlc
115	keuntungan national logic competition	benefit nlc
116	benefit nlc	benefit nlc
117	benefit national logic competition	benefit nlc
118	kenapa harus ikut npc	benefit npc
119	kenapa harus ikut national programming contest	benefit npc
120	mengapa harus ikut npc	benefit npc
121	mengapa harus ikut national programming contest	benefit npc
122	apa yang menarik dari npc	benefit npc
123	apa yang menarik dari national programming contest	benefit npc
124	apa keuntungan dari npc	benefit npc
125	apa keuntungan dari national programming contest	benefit npc
126	apa untungnya ikut npc	benefit npc
127	apa untungnya ikut national programming contest	benefit npc

No.	Dokumen	Kelas
128	mengapa npc menguntungkan bagi saya	benefit npc
129	mengapa national programming contest menguntungkan bagi saya	benefit npc
130	apa yang bisa saya dapatkan dari npc	benefit npc
131	apa yang bisa saya dapatkan dari national programming contest	benefit npc
132	jelaskan pada saya apa keuntungan mengikuti npc	benefit npc
133	jelaskan pada saya apa keuntungan mengikuti national programming contest	benefit npc
134	jelaskan mengapa saya harus ikut npc	benefit npc
135	jelaskan mengapa saya harus ikut national programming contest	benefit npc
136	terangkan pada saya benefit mengikuti npc	benefit npc
137	terangkan pada saya benefit mengikuti national programming contest	benefit npc
138	keuntungan npc	benefit npc
139	keuntungan national programming contest	benefit npc
140	benefit npc	benefit npc
141	benefit national programming contest	benefit npc
142	bagaimana sih contoh soal nlc	soal nlc
143	bagaimana sih contoh soal national programming contest	soal nlc
144	kasih contoh soal nlc dong	soal nlc
145	kasih contoh soal national logic competition dong	soal nlc
146	kisi-kisi soal nlc	soal nlc
147	kisi-kisi soal national logic competition	soal nlc

No.	Dokumen	Kelas
148	kisi kisi soal nlc	soal nlc
149	kisi kisi soal national logic competition	soal nlc
150	kasih contoh kisi-kisi soal nlc dong	soal nlc
151	kasih contoh kisi-kisi soal national logic competition dong	soal nlc
152	apa aja yang ditanyakan di nlc	soal nlc
153	apa aja yang ditanyakan di national logic competition	soal nlc
154	bagaimana sih bentuk soal nlc	soal nlc
155	bagaimana sih bentuk soal national logic competition	soal nlc
156	jenis soal nlc	soal nlc
157	jenis soal national logic competition	soal nlc
158	Jenis-jenis soal nlc	soal nlc
159	Jenis-jenis soal national logic competition	soal nlc
160	Soal-soal nlc	soal nlc
161	Soal-soal national logic competition	soal nlc
162	soal nlc	soal nlc
163	soal national logic competition	soal nlc
164	saya ingin ikut npc tapi tidak bisa programming	cannot code yet
165	saya ingin ikut national programming contest tapi tidak bisa programming	cannot code yet
166	ingin ikut npc belum bisa programming	cannot code yet
167	ingin ikut national programming contest belum bisa programming	cannot code yet
168	belum mengerti bahasa pemrograman	cannot code yet

No.	Dokumen	Kelas
169	saya belum pernah programming	cannot code yet
170	tidak mengerti programming	cannot code yet
171	apakah hanya yang bisa programming saya yang bisa ikut	cannot code yet
172	tidak mengerti bahasa pemrograman tapi ingin ikut national programming contest npc	cannot code yet
173	apakah nanti disediakan tutorial	cannot code yet
174	apakah nanti diajarkan programming oleh schematics npc	cannot code yet
175	bagaimana jika belum bisa melakukan pemrograman	cannot code yet
176	apakah harus bisa programming baru boleh ikut npc	cannot code yet
177	apakah harus bisa programming baru boleh ikut national programming contest	cannot code yet
178	apakah newbie bisa ikut npc	cannot code yet
179	apakah newbie bisa ikutan national programming contest	cannot code yet
180	apakah npc bisa diikuti yang baru belajar pemrograman	cannot code yet
181	apakah national programming contest bisa diikuti yang baru belajar pemrograman	cannot code yet
182	baru belajar programming	cannot code yet
183	baru belajar bahasa	cannot code yet
184	baru belajar pemrograman	cannot code yet
185	baru belajar coding	cannot code yet
186	belum bisa coding	cannot code yet
187	belum pernah coding sebelumnya	cannot code yet

No.	Dokumen	Kelas
188	saya belum pernah coding sebelumnya apakah bisa ikut npc	cannot code yet
189	saya belum jago pemrograman apakah saya bisa ikut npc	cannot code yet
190	apakah saya bisa ikut national programming contest padahal saya belum pernah coding sebelumnya	cannot code yet
191	apakah national programming contest hanya untuk yang jago coding saja	cannot code yet
192	bagaimana jika saya belum perna coding sama sekali	cannot code yet
193	apakah saya punya kesempatan menang jika belum bisa coding	cannot code yet
194	apakah lombanya npc susah	cannot code yet
195	apakah lomba national programming contest susah	cannot code yet
196	kapan nlc diadakan	where when nlc
197	kapan national logic competition diadakan	where when nlc
198	di mana nlc diadakan	where when nlc
199	di mana national logic competition diadakan	where when nlc
200	kapan dan di mana nlc diadakan	where when nlc
201	kapan dan di mana national logic competition diadakan	where when nlc
202	kapan nlc dihelat	where when nlc
203	di mana nlc dihelat	where when nlc
204	di mana national logic competition dihelat	where when nlc
205	apakah nlc diadakan di its	where when nlc

No.	Dokumen	Kelas
206	apakah national logic competition diadakan di its	where when nlc
207	apakah nlc di adakan di teknik informatika its	where when nlc
208	apakah national logic competition diadakan di teknik informatika its	where when nlc
209	apakah national logic competition diadakan di kota saya	where when nlc
210	apakah national logic competition diadakan di kota surabaya	where when nlc
211	apakah national logic competition diadakan di kota jakarta	where when nlc
212	apakah national logic competition diadakan di kota medan	where when nlc
213	apakah national logic competition diadakan di kota padang	where when nlc
214	apakah national logic competition diadakan di kota makassar	where when nlc
215	waktu nlc diadakan	where when nlc
216	waktu national logic competition diadakan	where when nlc
217	tanggal berapa nlc diadakan	where when nlc
218	tanggal berapa national logic competition diadakan	where when nlc
219	hari apa nlc mulainya	where when nlc
220	hari apa nlc mulainya	where when nlc
221	pada hari apa nlc diadakan	where when nlc
222	pada hati apa national logic competition dihelat	where when nlc
223	kapan npc diadakan	where when npc

No.	Dokumen	Kelas
224	kapan national programming contest diadakan	where when npc
225	di mana npc diadakan	where when npc
226	di mana national programming contest diadakan	where when npc
227	kapan dan di mana npc diadakan	where when npc
228	kapan dan di mana national programming contest diadakan	where when npc
229	kapan npc dihelat	where when npc
230	di mana npc dihelat	where when npc
231	di mana national programming contest dihelat	where when npc
232	apakah npc diadakan di its	where when npc
233	apakah national programming contest diadakan di its	where when npc
234	apakah npc di adakan di teknik informatika its	where when npc
235	apakah national programming contest diadakan di teknik informatika its	where when npc
236	apakah national programming contest diadakan di kota saya	where when npc
237	apakah national programming contest diadakan di kota surabaya	where when npc
238	apakah national programming contest diadakan di kota jakarta	where when npc
239	apakah national programming contest diadakan di kota medan	where when npc
240	apakah national programming contest diadakan di kota padang	where when npc

No.	Dokumen	Kelas
241	apakah national programming contest diadakan di kota makassar	where when npc
242	waktu npc diadakan	where when npc
243	waktu national programming contest diadakan	where when npc
244	tanggal berapa npc diadakan	where when npc
245	tanggal berapa national programming contest diadakan	where when npc
246	hari apa national programming contest mulainya	where when npc
247	hari apa npc mulainya	where when npc
248	pada hari apa npc diadakan	where when npc
249	pada hari apa national programming contest dihelat	where when npc
250	bagaimana cara daftar nlc	pendaftaran nlc npc
251	bagaimana cara daftar npc	pendaftaran nlc npc
252	bagaimana cara daftar national logic competition	pendaftaran nlc npc
253	bagaimana cara daftar national programming contest	pendaftaran nlc npc
254	apa yang harus saya lakukan jika ingin mendaftar	pendaftaran nlc npc
255	daftar nlc	pendaftaran nlc npc
256	daftar npc	pendaftaran nlc npc
257	mendaftar national logic competition	pendaftaran nlc npc

No.	Dokumen	Kelas
258	mendaftar national programming contest	pendaftaran nlc npc
259	apa yang bisa saya lakukan untuk mendaftar	pendaftaran nlc npc
260	informasi pendaftaran	pendaftaran nlc npc
261	boleh saya minta informasi pendaftarannya	pendaftaran nlc npc
262	di mana saya bisa mendaftar untuk ikut	pendaftaran nlc npc
263	saya ingin ikut lombanya	pendaftaran nlc npc
264	saya ingin ikut	pendaftaran nlc npc
265	Boleh saya minta form pendaftarannya npc	pendaftaran nlc npc
266	boleh saya minta form pendaftarannya nlc	pendaftaran nlc npc
267	boleh saya minta form pendaftarannya nlc npc	pendaftaran nlc npc
268	prosedur pendaftaran nlc	pendaftaran nlc npc
269	prosedur pendaftaran npc	pendaftaran nlc npc
270	saya ingin minta pendaftaran nlc	pendaftaran nlc npc
271	saya ingin minta prosedur pendaftaran npc	pendaftaran nlc npc
272	bagaimana pendaftarannya national logic competition	pendaftaran nlc npc
273	sampai kapan pendaftaran nlc ditutup	timeline nlc

No.	Dokumen	Kelas
274	sampai kapan pendaftaran national logic competition ditutup	timeline nlc
275	boleh saya minta jadwal pendaftaran nlc	timeline nlc
276	boleh saya minta jadwal pendaftaran national logic competition	timeline nlc
277	jadwal nlc	timeline nlc
278	jadwal national logic competition	timeline nlc
279	pendaftarannya nlc sampai kapan ya	timeline nlc
280	pendaftarannya national logic competition sampai kapan ya	timeline nlc
281	kapan pendaftarannya nlc ditutup	timeline nlc
282	kapan pendaftarannya nlc dibuka	timeline nlc
283	saya minta timelne nlc dong	timeline nlc
284	saya minta timeline national logic competition dong	timeline nlc
285	timeline nlc	timeline nlc
286	timeline national logic competition	timeline nlc
287	boleh saya minta timeline national logic competition	timeline nlc
288	boleh saya minta timeline nlc	timeline nlc
289	sampai kapan pendaftaran npc ditutup	timeline npc
290	sampai kapan pendaftaran national programming contest ditutup	timeline npc
291	boleh saya minta jadwal pendaftaran npc	timeline npc
292	boleh saya minta jadwal pendaftaran national programming contest	timeline npc
293	jadwal npc	timeline npc
294	jadwal national programming contest	timeline npc

No.	Dokumen	Kelas
295	pendaftarannya npc sampai kapan ya	timeline npc
296	pendaftarannya national programming contest sampai kapan ya	timeline npc
297	kapan pendaftarannya npc ditutup	timeline npc
298	kapan pendaftarannya npc dibuka	timeline npc
299	saya minta timelne npc dong	timeline npc
300	saya minta timeline national programming contest dong	timeline npc
301	timeline npc	timeline npc
302	timeline national programming contest	timeline npc
303	boleh saya minta timeline national programming contest	timeline npc
304	boleh saya minta timeline npc	timeline npc
305	berapa biaya pendaftaran nlc	biaya nlc
306	boleh saya tanya harga pendaftaran national logic competition	biaya nlc
307	berapa biaya pendaftaran national logic competition	biaya nlc
308	berapa rupiah daftar nlc	biaya nlc
309	berapa rupiah daftar national logic competition	biaya nlc
310	harganya nlc berapa ya	biaya nlc
311	harganya national logic competition berapa ya	biaya nlc
312	berapa yang harus dibayar untuk daftar national logic competition	biaya nlc
313	berapa yang harus dibayar untuk daftar nlc	biaya nlc
314	biaya nlc	biaya nlc

No.	Dokumen	Kelas
315	biaya national logic competition	biaya nlc
316	harga nlc	biaya nlc
317	harga national logic competition	biaya nlc
318	berapa ratus ribu uang yang harus dibayarkan untuk nlc	biaya nlc
319	berapa ratus ribu uang yang harus dibayarkan untuk national logic competition	biaya nlc
320	berapa biaya pendaftaran npc	biaya npc
321	boleh saya tanya harga pendaftaran national programming contest	biaya npc
322	berapa biaya pendaftaran national programming contest	biaya npc
323	berapa rupiah daftar npc	biaya npc
324	berapa rupiah daftar npc	biaya npc
325	harganya npc berapa ya	biaya npc
326	harganya national programming contest berapa ya	biaya npc
327	berapa yang harus dibayar untuk daftar national programming contest	biaya npc
328	berapa yang harus dibayar untuk daftar npc	biaya npc
329	biaya npc	biaya npc
330	biaya national programming contest	biaya npc
331	harga npc	biaya npc
332	harga national programming contest	biaya npc
333	berapa ratus ribu uang yang harus dibayarkan untuk npc	biaya npc

No.	Dokumen	Kelas
334	berapa ratus ribu uang yang harus dibayarkan untuk national programming contest	biaya npc
335	saya sudah daftar tapi pendaftaran saya berlum terverifikasi kenapa ya	belum verifikasi
336	belum terverifikasi	belum verifikasi
337	verifikasi saya belum berhasil	belum verifikasi
338	mengapa verifikasi pendaftaran nlc saya belum juga	belum verifikasi
339	mengapa verifikasi pendaftaran national logic competition saya belum	belum verifikasi
340	verifikasinya berapa lama	belum verifikasi
341	apakah verifikasinya membutuhkan waktu yang lama	belum verifikasi
342	mengapa kelompok saya belum terverifikasi	belum verifikasi
343	mengapa kelompok saya belum terverifikasi padahal saya sudah bayar sejak lama	belum verifikasi
344	apakah saya sudah terverifikasi	belum verifikasi
345	apakah transfer uang saya sudah diterima	belum verifikasi
346	apakah pembayaran transfer saya sudah berhasil	belum verifikasi
347	saya sudah transfer sejak pagi apakah sudah diterima	belum verifikasi
348	berapa lama kira-kira transfer uang untuk pendaftaran saya diverifikasi	belum verifikasi
349	verifikasi pendaftaran	belum verifikasi
350	verifikasi	belum verifikasi

No.	Dokumen	Kelas
351	apakah national logic competition bisa pindah region	pindah region nlc
352	apakah nlc bisa pindah region	pindah region nlc
353	saya mendaftar nlc online apa perlu untuk memilih region	pindah region nlc
354	say amendaftar nlc secara online apakah bisa pindah offline dengan memilih region	pindah region nlc
355	aturan region nlc	pindah region nlc
356	aturan region national logic competition	pindah region nlc
357	adakah aturan region nlc	pindah region nlc
358	adakah aturan region national logic competition	pindah region nlc
359	saya mau pindah region nlc	pindah region nlc
360	saya mau pindah region national logic competition	pindah region nlc
361	bagaimana caranya pindah region di nlc	pindah region nlc
362	bagaimana caranya pindah region national logic competition	pindah region nlc
363	jumlah kelompok nlc	jumlah kelompok
364	jumlah kelompok national logic competition	jumlah kelompok
365	bolehkah kelompok national logic competition dua orang	jumlah kelompok
366	bolehkan kelompok national logic competition lebih dari satu orang	jumlah kelompok
367	kelompok nlc	jumlah kelompok
368	grup nlc	jumlah kelompok
369	grup national logic competition	jumlah kelompok

No.	Dokumen	Kelas
370	saya ingin bentuk kelompok lebih dari boleh tidak	jumlah kelompok
371	apakah boleh kelompok national logic competition kurang dari tiga orang	jumlah kelompok
372	apakah boleh nlc kelompoknya dari sekolah yang berbeda	jumlah kelompok
373	national logic competition apakah kelompoknya harus dari sekolah yang sama	jumlah kelompok
374	nlc apakah kelompoknya dari sekolah yang sama'	jumlah kelompok
375	grup dari sekolah yang berbeda	jumlah kelompok
376	sekolah yang berbeda	jumlah kelompok
377	berapa jumlah kelompok npc	jumlah kelompok
378	bolehkan national programming contest lebih dari satu orang	jumlah kelompok
379	apakah npc boleh ikut lebih dari satu orang	jumlah kelompok
380	national programming contest perorangan atau bagaimana	jumlah kelompok
381	npc perorangan atau bagaimana	jumlah kelompok
382	npc hanya satu orang saja ya	jumlah kelompok
383	apakah boleh lebih dari satu	jumlah kelompok
384	apakah boleh kurang dari tiga	jumlah kelompok
385	mengapa sertifikat saya tidak kunjung datang ya	sertifikat tidak datang
386	sudah seminggu saya ikut nlc tetapi sertifikat saya belum dikirim juga	sertifikat tidak datang
387	sudah lama saya ikut npc tetapi sertifikat saya belum datang juga	sertifikat tidak datang

No.	Dokumen	Kelas
388	mengapa sertifikat lama sekali	sertifikat tidak datang
389	apakah sertifikat akan segera dikirim	sertifikat tidak datang
390	sertifikat	sertifikat tidak datang
391	sertifikat saya mana	sertifikat tidak datang
392	kapan sertifikat saya datang	sertifikat tidak datang
393	kapan sertifikat saya dan kelompok saya datang	sertifikat tidak datang
394	sertifikat national logic competition	sertifikat tidak datang
395	sertifikat national programming contest	sertifikat tidak datang
396	saya sudah ikut lomba dan tereleminas pada babak mengapa sertifikat saya tidak kunjung datang	sertifikat tidak datang
397	mohon sertifikat saya dipercepat karena saya ada sekian	sertifikat tidak datang
398	mohon sertifikat kelompok saya segera dikirim	sertifikat tidak datang
399	apakah nst masih ada tiket	tiket nst reeva
400	apakah tiket national seminar of technology masih ada	tiket nst reeva
401	apakah tiket reeva masih ada	tiket nst reeva
402	apakah tiket revolutionary entertaiment and expo with various arts	tiket nst reeva
403	apakah tiket reeva masih ada	tiket nst reeva
404	tiket sudah sold out belum	tiket nst reeva

No.	Dokumen	Kelas
405	apakah tiket nst dan reeva sudah sold out	tiket nst reeva
406	saya dengar tiketnya sudah sold out ya	tiket nst reeva
407	apakah benar tiket sudah terjual semua	tiket nst reeva
408	tiket nst sudah terjual semua apa belum	tiket nst reeva
409	tiket reeva sudah terjual semua atau belum	tiket nst reeva
410	tiket national seminar of technology sudah terjual semua belum sih	tiket nst reeva
411	tiket revolutionary expo and expo with various arts sudah terjual semua belum	tiket nst reeva
412	apakah masih ada tiketnya	tiket nst reeva
413	mau cek tiket reeva masih ada atau belum	tiket nst reeva
414	harga tiket nst	tiket nst reeva
415	harga tiket reeva	tiket nst reeva
416	harga tiket national seminar of technology schematics	tiket nst reeva
417	harga tiket reeva	tiket nst reeva
418	berapa harga tiketnya nst	tiket nst reeva
419	berapa harga tiketnya reeva sekarang	tiket nst reeva
420	info harga tiket nst dan reeva dong	tiket nst reeva
421	info harga tiket nst dong	tiket nst reeva
422	apakah tiketnya masih ada	pengumuman nlc npc
423	kapan pengumuman nlc	pengumuman nlc npc
424	kapan pengumuman npc	pengumuman nlc npc
425	kapan pengumuman national logic competition	pengumuman nlc npc

No.	Dokumen	Kelas
426	kapan pengumuman national programming contest	pengumuman nlc npc
427	info pengumuman hasil dong	pengumuman nlc npc
428	pengumuman nlc kapan ya	pengumuman nlc npc
429	kalau boleh tau kapan ya pengumuman npc	pengumuman nlc npc
430	npc pengumumannya kapan ya kalau boleh tau	pengumuman nlc npc
431	apakah sudah ada informasi pengumuman nlc	pengumuman nlc npc
432	apakah sudah ada informasi pengumuman national logic competition	pengumuman nlc npc
433	apakah sudah ada informasi pengumuman national programming contest	pengumuman nlc npc
434	apa benar pengumuman nlc minggu depan	pengumuman nlc npc
435	apa benar pengumuman npc minggu depan	pengumuman nlc npc
436	di mana saya bisa tau pengumuman nlc ya	pengumuman nlc npc
437	di mana saya bisa tau pengumuman npc ya	pengumuman nlc npc
438	informasi pengumuman bisa diakses di mana ya	pengumuman nlc npc
439	informasi pengumuman nlc bisa diakses di mana ya	pengumuman nlc npc
440	informasi pengumuman bisa didapat di mana ya	pengumuman nlc npc

No.	Dokumen	Kelas
441	di mana saya bisa dapat informasi pengumuman nlc	pengumuman nlc npc
442	di mana saya bisa dapat informasi pengumuman npc ya	pengumuman nlc npc
443	saya lihat di web pengumuman nlc belum ada	pengumuman nlc npc
444	saya lihat di web belum ada pengumuman npc	pengumuman nlc npc
445	apakah ketika lomba nanti harus pakai seragam	aturan seragam
446	pake seragam tidak	aturan seragam
447	apa harus menggunakan seragam sekolah	aturan seragam
448	baju apa yang digunakan saat lomba nanti	aturan seragam
449	apakah boleh menggunakan baju bebas	aturan seragam
450	baju apa yang dikenakan saat nlc	aturan seragam
451	baju apa yang dikenakan saat npc	aturan seragam
452	apakah harus menggunakan seragam putih abu-abu untuk ikut national logic competition	aturan seragam
453	bagaimana jika saya menggunakan baju bebas	aturan seragam
454	apakah boleh menggunakan baju bebas	aturan seragam
455	baju bebas	aturan seragam
456	seragam sekolah	aturan seragam
457	apakah harus menggunakan seragam sekolah	aturan seragam
458	bagaimana jika saya tidak punya seragam sekolah	aturan seragam
459	bagaimana jika saya kehilangan baju saya	aturan seragam

No.	Dokumen	Kelas
460	harus menggunakan seragam	aturan seragam
461	harus menggunakan baju bebas	aturan seragam
462	saya hanya punya baju bebas	aturan seragam

BIODATA PENULIS

Penulis, bernama lengkap Irfan Hanif, lahir di Jakarta, 16 Agustus 1996. Penulis merupakan mahasiswa Teknik Informatika ITS 2014. Selain aktif dalam berbagai kegiatan kemahasiswaan, penulis juga adalah seorang finalis berbagai kompetisi teknologi informasi dan administrator Mobile Innovation Studio (MIS Informatika ITS). Diantara berbagai prestasi yang pernah diraih adalah Juara 2 tingkat nasional MAGE Teknik Komputer ITS bidang aplikasi, finalis Go-Hackathon Go-Jek 2017, finalis GEMASTIK 10 bidang Pengembangan Bisnis Perangkat Lunak, dan finalis Hackathon Arkavidia ITB 2018. Penulis juga tercatat sebagai Ketua Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS periode kepengurusan 2016/2017. Pada tahun-tahun awal masa kuliah, penulis pernah menjabat sebagai staf HMTC Departemen Kaderisasi dan Pemetaan, staf BEM Fakultas Teknologi Informasi (FTIf) Departemen *External Affair*, dan Pemandu Fakultas Teknologi Informasi.

Dalam menyelesaikan pendidikan sarjana, penulis mengambil bidang minat Komputasi Cerdas dan Visi (KCV), tidak lain karena ketertarikannya dalam bidang *machine learning* dan kecerdasan buatan pada umumnya. Sebagai Administrator MIS, penulis juga termasuk dalam tim penelitian Sistem Kehadiran Mahasiswa Informatika (Sikemas) yang berfokus pada pengembangan sistem *face recognition*.